NXP Semiconductors
Reference Manual

Document identifier: MCXCP048M072F20RM
Rev. 2, 2026-07-29

MCX C151/161/162 Reference Manual

Supports MCX C151, MCX C161, and MCX C162

Contents

Chapter 1 About This Manual..... 10
1.1 Audience.....10
1.2 Organization..... 10
1.3 Module descriptions.....10
1.4 Register descriptions..... 13
1.5 Conventions.....15
1.6 Editorial changes..... 17

Chapter 2 Introduction..... 18
2.1 Overview.....18
2.2 Target applications..... 18
2.3 Block diagram.....18
2.4 Features..... 19
2.5 Functional overview.....21

Chapter 3 Core Overview..... 27
3.1 Introduction.....27
3.2 Cortex-M23 core.....27
3.3 Nested Vectored Interrupt Controller (NVIC).....28
3.4 System memory map.....32
3.5 Peripheral bridge 34

Chapter 4 Bus and Memory Architecture.....38
4.1 Bus and memory architecture.....38
4.2 System bus priority and arbitration..... 39
4.3 SRAM configuration.....39
4.4 Flash subsystem.....40
4.5 IFR configuration..... 42

Chapter 5 Flash Memory Module (FMU)..... 43
5.1 Chip-specific FMU information..... 43
5.2 Overview.....43
5.3 Functional description.....45
5.4 External signals..... 60
5.5 Initialization.....60
5.6 Memory map and registers.....60
5.7 Glossary..... 72

Chapter 6 Flash Memory Controller (FMC).....74
6.1 Chip-specific FMC information..... 74
6.2 Overview.....74
6.3 Functional description.....75
6.4 External signals..... 76
6.5 Initialization and application information.....76

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Chapter 7 Error Injection Module (EIM)..... 78
7.1 Chip-specific EIM information.....78
7.2 Overview.....78
7.3 Functional description.....80
7.4 Initialization.....80
7.5 EIM register descriptions.....80

Chapter 8 Error Recording Module (ERM) 88
8.1 Chip-specific Error Recording information.....88
8.2 Overview.....88
8.3 Functional description.....89
8.4 Initialization.....90
8.5 ERM register descriptions..... 91

Chapter 9 Signal Multiplexing..... 99
9.1 Introduction.....99
9.2 Signal multiplexing and pin assignments.....99

Chapter 10 Port Control (PORT).....107
10.1 Chip-specific PORT information..... 107
10.2 Overview.....108
10.3 Functional description.....108
10.4 Initialization.....110
10.5 Application information..... 110
10.6 Memory map and register definition..... 110

Chapter 11 General Purpose Input/Output (GPIO).....182
11.1 Chip-specific GPIO information..... 182
11.2 Overview.....182
11.3 Functional description.....183
11.4 External signals 185
11.5 Initialization.....185
11.6 Application information..... 186
11.7 Memory map and register definition..... 186

Chapter 12 Input Multiplexing (INPUTMUX)..... 201
12.1 Chip-specific INPUTMUX information..... 201
12.2 Overview.....201
12.3 TRIG_IN Filter.....202
12.4 Functional description.....203
12.5 External signals..... 203
12.6 Memory map and register definition..... 203

Chapter 13 System Controller (SYSCON)..... 273
13.1 Chip-specific SYSCON information.....273
13.2 Overview.....273
13.3 Functional description.....274
13.4 Memory map and register definition..... 274
13.5 Application information..... 361

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Chapter 14 Enhanced DMA Controller (eDMA3)..... 362
14.1 Chip-specific eDMA information..... 362
14.2 Overview.....364
14.3 Functional description.....367
14.4 External signals..... 372
14.5 Initialization.....372
14.6 Memory map and register definition..... 384

Chapter 15 Wakeup Unit (WUU).....424
15.1 Chip-specific WUU information.....424
15.2 Overview.....426
15.3 Functional description.....428
15.4 External signals..... 429
15.5 Initialization.....429
15.6 Application information..... 430
15.7 Memory map and register definition..... 430

Chapter 16 Core Mode Controller (CMC)..... 465
16.1 Chip-specific CMC information.....465
16.2 Overview.....465
16.3 Functional description.....465
16.4 External signals..... 472
16.5 Initialization.....472
16.6 Application information..... 472
16.7 Memory map and register descriptions..... 473

Chapter 17 Reset.....499
17.1 Reset sources.....499
17.2 Power reset sources.....499
17.3 External reset sources.....500
17.4 Internal reset sources..... 501
17.5 Reset type..... 502

Chapter 18 Lifecycle States..... 505
18.1 Overview.....505
18.2 Lifecycle states..... 505
18.3 Lifecycle state transitions..... 505
18.4 Customer lifecycle states.....506

Chapter 19 Bootloader.....509
19.1 Overview.....509
19.2 Boot flow.....509
19.3 In-System Programming (ISP).....521
19.4 Debug Mailbox.....551
19.5 Run bootloader API..... 556

Chapter 20 Clocking..... 558
20.1 Introduction.....558
20.2 Configure main clock and system clock..... 558

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

20.3 Clock generation....558

Chapter 21 System Clock Generator (SCG)....568
21.1 Chip-specific SCG information....568
21.2 Overview....568
21.3 Functional description....570
21.4 External signals....572
21.5 Initialization....572
21.6 Application information....573
21.7 Memory map and register definition....574

Chapter 22 Power Management....599
22.1 Introduction....599
22.2 Power domains....599
22.3 Power modes....601
22.4 Module operation in low power modes....602
22.5 Power optimization....604

Chapter 23 VBAT....607
23.1 Chip-specific VBAT information....607
23.2 Overview....607
23.3 Functional description....608
23.4 External signals....609
23.5 Initialization....609
23.6 Memory map and register definition....609

Chapter 24 System Power Control (SPC)....626
24.1 Chip-specific SPC information....626
24.2 Overview....627
24.3 Functional description....628
24.4 External signals....633
24.5 Initialization....633
24.6 Application information....633
24.7 SPC register descriptions....633

Chapter 25 Standard Counter/Timers (CTIMER)....665
25.1 Chip-specific CTIMER information....665
25.2 Overview....666
25.3 Functional description....668
25.4 External signals....670
25.5 Initialization....671
25.6 Application information....671
25.7 CTIMER register descriptions....671

Chapter 26 Windowed Watchdog Timer (WWDT)....691
26.1 Chip-specific WWDT information....691
26.2 Overview....691
26.3 Functional description....693
26.4 External signals....695

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

26.5 Initialization....696
26.6 Memory map and register definition....696

Chapter 27 Real Time Clock (RTC)....704
27.1 Chip-specific RTC information....704
27.2 Overview....704
27.3 Functional description....705
27.4 Initialization....707
27.5 Register descriptions....707

Chapter 28 Enhanced Flex Pulse Width Modulator (eFlexPWM)....719
28.1 Chip-specific PWM information....719
28.2 Overview....719
28.3 Functional description....721
28.4 External Signals....751
28.5 eFlexPWM register descriptions....752

Chapter 29 Frequency Measurement (FREQME)....814
29.1 Chip-specific FREQME information....814
29.2 Overview....815
29.3 Functional description....815
29.4 External signals....817
29.5 Initialization....817
29.6 Memory map and register definition....817

Chapter 30 Low-power Timer (LPTMR)....826
30.1 Chip-specific LPTMR information....826
30.2 Overview....827
30.3 Functional description....828
30.4 External signals....830
30.5 Initialization....830
30.6 Application information....831
30.7 Memory map and register definition....831

Chapter 31 Low Power Inter-Integrated Circuit (LPI2C)....839
31.1 Chip-specific LPI2C information....839
31.2 Overview....839
31.3 Functional description....841
31.4 External signals....852
31.5 Initialization....853
31.6 Application information....854
31.7 Memory map and registers....854

Chapter 32 Low Power Universal Asynchronous Receiver/Transmitter (LPUART)....908
32.1 Chip-specific LPUART information....908
32.2 Overview....908
32.3 Functional description....911
32.4 External signals....925

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

32.5 Initialization....925
32.6 LPUART register descriptions....926

Chapter 33 Low Power Serial Peripheral Interface (LPSPI)....964
33.1 Chip-specific LPSPI information....964
33.2 Overview....964
33.3 Functional description....966
33.4 External signals....978
33.5 Initialization....979
33.6 Memory map and registers....979

Chapter 34 Analog-to-Digital Converter (ADC)....1013
34.1 Chip-specific ADC information....1013
34.2 Overview....1015
34.3 Functional description....1017
34.4 External signals....1027
34.5 Initialization....1028
34.6 ADC register descriptions....1030

Chapter 35 Operational Amplifier (OPAMP)....1070
35.1 Chip-specific OPAMP information....1070
35.2 Overview....1071
35.3 Functional description....1072
35.4 External Signals....1075
35.5 Initialization....1075
35.6 Application Information....1075
35.7 Memory map and register definition....1076

Chapter 36 Low Power Comparator (LPCMP)....1083
36.1 Chip-specific LPCMP information....1083
36.2 Overview....1084
36.3 Functional Description....1086
36.4 External signal descriptions....1099
36.5 Initialization....1099
36.6 Application information....1100
36.7 LPCMP register descriptions....1101

Chapter 37 Debug....1122
37.1 Introduction....1122
37.2 Debug component....1123
37.3 Debug port connectivity....1123
37.4 Debug ROM tables....1125

Chapter 38 Debug Mailbox (DBGMB)....1126
38.1 Overview....1126
38.2 Functional description....1126
38.3 External signals....1126
38.4 Memory map and register definition....1127

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Chapter 39 Cyclic Redundancy Check (CRC)..... 1133
39.1 Chip-specific CRC information..... 1133
39.2 Overview..... 1133
39.3 Functional description..... 1134
39.4 External signals..... 1136
39.5 Initialization..... 1137
39.6 Use cases..... 1137
39.7 Memory map and register descriptions..... 1139

Chapter 40 Memory Block Checker (MBC)..... 1144
40.1 Chip-specific MBC information..... 1144
40.2 Overview..... 1145
40.3 Functional description..... 1146
40.4 External signals..... 1148
40.5 Register descriptions..... 1148

Chapter 41 True Random Generator (TRNG)..... 1180
41.1 Chip-specific TRNG information..... 1180
41.2 Overview..... 1180
41.3 Functional Description..... 1181
41.4 Register Interface Usage..... 1181
41.5 Operation..... 1182
41.6 Other Applications..... 1185
41.7 Memory Map and Registers..... 1185

Appendix A Release notes..... 1235
A.1 About This Manual changes..... 1235
A.2 Introduction changes..... 1235
A.3 Core overview changes..... 1235
A.4 Bus and memory architecture chapter changes..... 1235
A.5 Flash Memory Module (FMU)..... 1235
A.6 Flash Memory Controller (FMC)..... 1235
A.7 Error Injection Module (EIM)..... 1235
A.8 Error Recording Module (ERM)..... 1236
A.9 Signal multiplexing chapter changes..... 1236
A.10 Port Control (PORT)..... 1236
A.11 General Purpose Input/Output (GPIO)..... 1236
A.12 Input Multiplexing (INPUTMUX)..... 1236
A.13 System Controller (SYSCON)..... 1237
A.14 Enhanced DMA Controller (eDMA3)..... 1237
A.15 Wakeup Unit (WUU)..... 1237
A.16 Core Mode Controller (CMC)..... 1237
A.17 Reset changes..... 1238
A.18 Lifecycle States changes..... 1238
A.19 Bootloader changes..... 1238
A.20 Clocking chapter changes..... 1238
A.21 System Clock Generator (SCG)..... 1238
A.22 Power Management chapter changes..... 1239
A.23 VBAT..... 1239
A.24 System Power Control (SPC)..... 1239

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

A.25 Standard Counter/Timers (CTIMER)....1239
A.26 Windowed Watchdog Timer (WWDT)....1240
A.27 Real Time Clock (RTC)....1240
A.28 Enhanced Flex Pulse Width Modulator (eFlexPWM)....1240
A.29 Frequency Measurement (FREQME)....1240
A.30 Low-power Timer (LPTMR)....1241
A.31 Low Power Inter-Integrated Circuit (LPI2C)....1241
A.32 Low Power Universal Asynchronous Receiver/Transmitter (LPUART)....1241
A.33 Low Power Serial Peripheral Interface (LPSPI)....1241
A.34 Analog-to-Digital Converter (ADC)....1241
A.35 Operational Amplifier (OPAMP)....1242
A.36 Low Power Comparator (LPCMP)....1242
A.37 Debug chapter changes....1242
A.38 DBGMB module changes....1242
A.39 Cyclic Redundancy Check (CRC)....1242
A.40 Memory Block Checker (MBC)....1243
A.41 True Random Generator (TRNG)....1243

Legal information....1245

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Chapter 1
About This Manual

1.1 Audience

This reference manual (RM) is intended for system software, hardware developers, and applications programmers who need to develop products using this chip. It assumes that its users understand operating systems, microprocessor system design, and basic principles of software and hardware.

1.2 Organization

This manual has two main sets of chapters.

- Chapters in the first set contain information that applies to all components on the chip.
- Chapters in the second set are organized into functional groupings that detail particular areas of functionality.
  - Examples of these groupings are clocking, timers, and communication interfaces.
  - Each grouping includes chapters that provide a technical description of individual modules.

1.2.1 Attachments

This manual includes key information in the files attached to it. For example, memory map and I/O details. Use the content in these attachments in conjunction with this manual's content.

NOTE

Select the paperclip icon on the left side of the PDF window to see the list of attachments.

1.3 Module descriptions

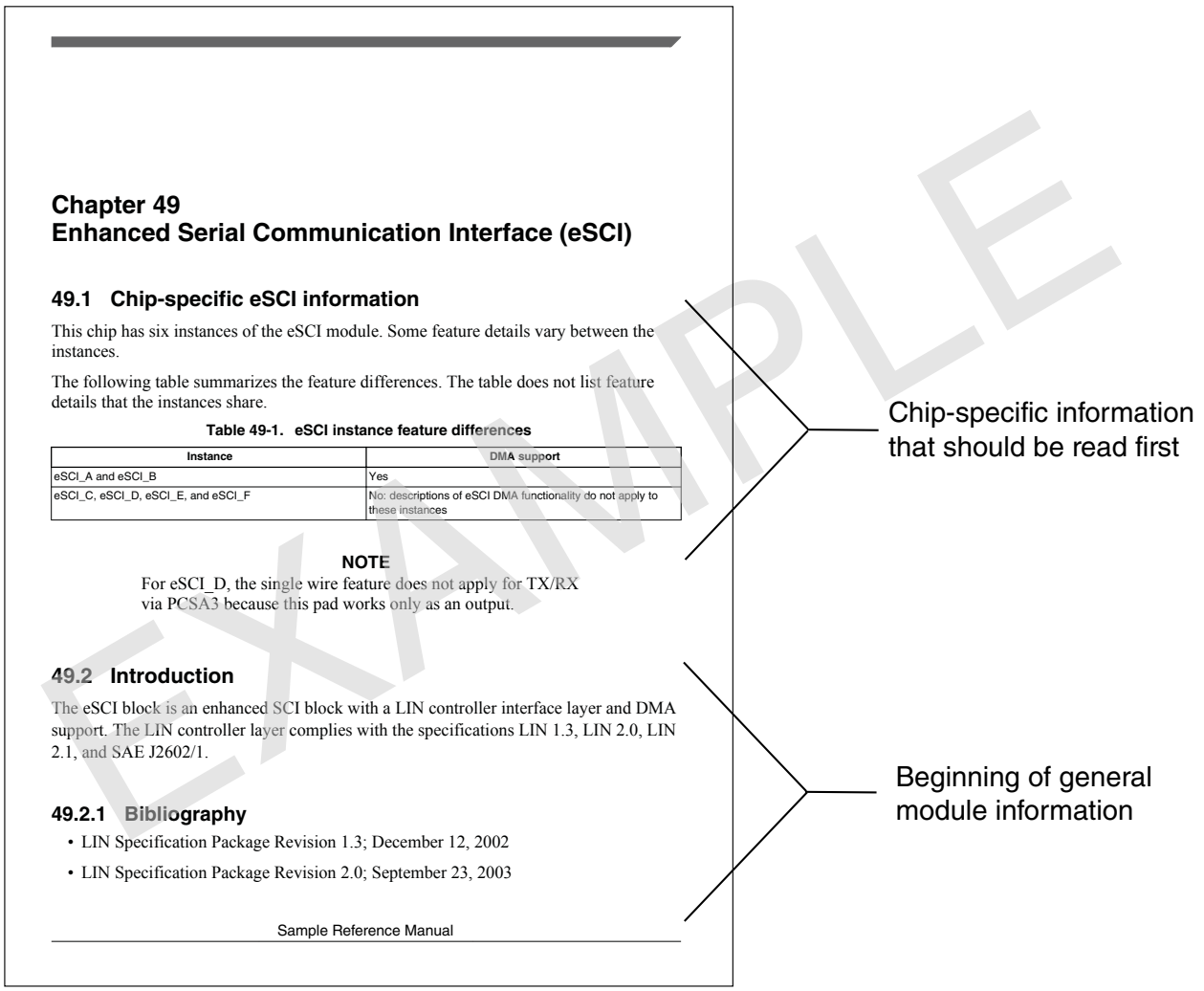
Each module chapter has two main parts:

- The first section, *chip-specific [module name]* information, provides details such as the number of module instances on the chip and connections between that module and the other ones. Read this section *first* because its content is crucial for understanding the information in the other sections of the chapter.
- The subsequent sections provide general information about the module, including its signals, registers, and functional description.

The following figure shows you an example of this demarcation.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.



Chapter 49
Enhanced Serial Communication Interface (eSCI)

49.1 Chip-specific eSCI information

This chip has six instances of the eSCI module. Some feature details vary between the instances.

The following table summarizes the feature differences. The table does not list feature details that the instances share.

Table 49-1. eSCI instance feature differences

Instance	DMA support
eSCI_A and eSCI_B	Yes
eSCI_C, eSCI_D, eSCI_E, and eSCI_F	No; descriptions of eSCI DMA functionality do not apply to these instances

NOTE
For eSCI_D, the single wire feature does not apply for TX/RX via PCSA3 because this pad works only as an output.

49.2 Introduction

The eSCI block is an enhanced SCI block with a LIN controller interface layer and DMA support. The LIN controller layer complies with the specifications LIN 1.3, LIN 2.0, LIN 2.1, and SAE J2602/1.

49.2.1 Bibliography

- LIN Specification Package Revision 1.3; December 12, 2002
- LIN Specification Package Revision 2.0; September 23, 2003

Sample Reference Manual

EXAMPLE

Figure 1. Example of chapter chip-specific information and general module information

1.3.1 Chip-specific information that clarifies content in the same chapter

The following figure shows an example of chip-specific information that clarifies general module information presented later in the chapter. In this case, the chip-specific register reset values supersede the reset values that appear in the register diagram.

All information provided in this document is subject to legal disclaimers.
© 2026 NXP B.V. All rights reserved.

Chapter 9
System Integration Unit Lite2 (SIUL2)

9.1 Chip-specific SIUL2 information

9.1.1 Feature configurations

In this device, the SIUL2_0 module instance does not support the following features described in the generic description:

- Interrupts
- DMA channels

9.1.2 Notes for IMCR

Out of reset, PA_00, PA_04, and PA_05 pads have JTAG input functionality selected by default. It should be disabled in the corresponding IMCR registers (IMCR61, IMCR60, and IMCR50 respectively) in order to use other functionality such as GPIO.

9.2 Introduction

9.2.1 Overview

The System Integration Unit Lite2 provides control over all the electrical pin controls and ports with 16 bits of bidirectional, general-purpose input and output signals. One of the most important functions of the SIUL2 is to enable the user to select the functions and electrical characteristics that appear on external device pins. It also controls the multiplexing of internal signals from one module to another and controls chip I/O. It supports as many as 32 external interrupts with trigger event configuration. The following figure is the block diagram of SIUL2 and its interfaces to other system components.

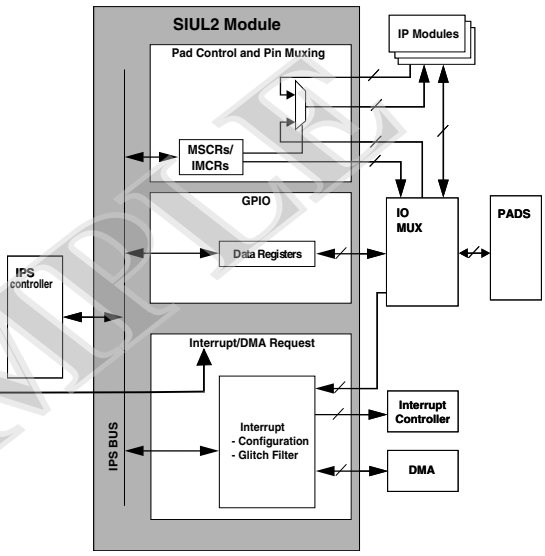


Figure 23. System Integration Unit Lite2 block diagram

This module provides dedicated pad control to general-purpose pads that can be configured as either inputs or outputs. The SIUL2 module provides registers that enable user software to read values from GPIO pads configured as inputs, and write values to GPIO pads configured as outputs:

- When configured as output, you can write to an internal register to control the state driven on the associated output pad.
- When configured as input, you can detect the state of the associated pad by reading the value from an internal register.
- When configured as input and output, the pad value can be read back, which can be used as a method of checking if the written value appeared on the pad.

To assist software development, GPIO data registers can be accessed using various mechanisms. These differing mechanisms allow support for port access or for bit manipulation without the need to use read-modify-write operations:

- Access to two 16-bit ports in one access
- Read/write access to a single bit
- A 16-bit port write with a bit mask, using single 32-bit access.

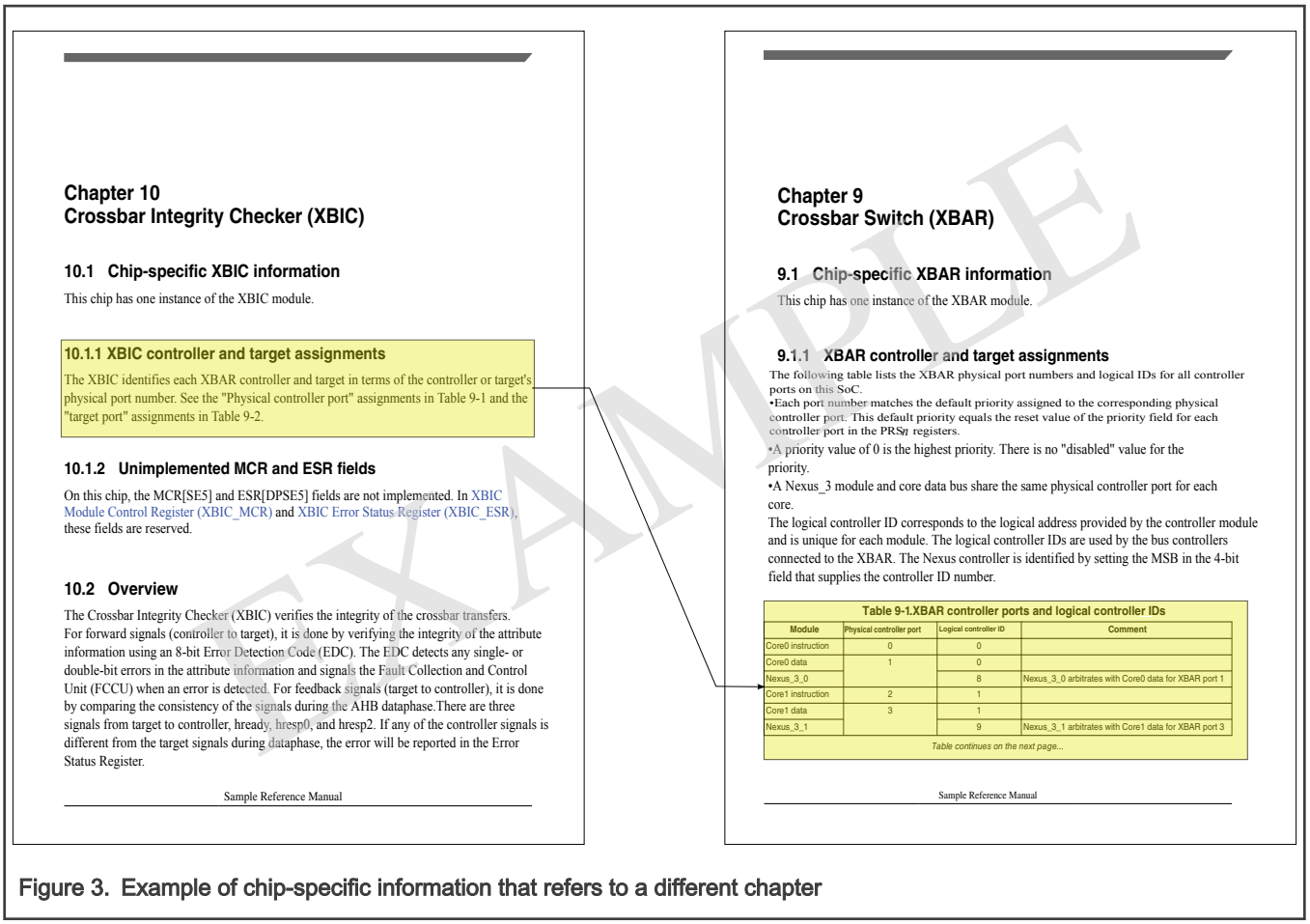
EXAMPLE

Figure 2. Example of chip-specific information that clarifies content in the same chapter

1.3.2 Chip-specific information that refers to a different chapter

Related chip-specific information may be provided in different chapters of the manual. The following figure shows an example of two such connected pieces of information. In this case, read both before you proceed.

© 2026 NXP B.V. All rights reserved.



Chapter 10
Crossbar Integrity Checker (XBIC)

10.1 Chip-specific XBIC information

This chip has one instance of the XBIC module.

10.1.1 XBIC controller and target assignments

The XBIC identifies each XBAR controller and target in terms of the controller or target's physical port number. See the "Physical controller port" assignments in Table 9-1 and the "target port" assignments in Table 9-2.

10.1.2 Unimplemented MCR and ESR fields

On this chip, the MCR[SE5] and ESR[DPSE5] fields are not implemented. In XBIC Module Control Register (XBIC_MCR) and XBIC Error Status Register (XBIC_ESR), these fields are reserved.

10.2 Overview

The Crossbar Integrity Checker (XBIC) verifies the integrity of the crossbar transfers. For forward signals (controller to target), it is done by verifying the integrity of the attribute information using an 8-bit Error Detection Code (EDC). The EDC detects any single- or double-bit errors in the attribute information and signals the Fault Collection and Control Unit (FCCU) when an error is detected. For feedback signals (target to controller), it is done by comparing the consistency of the signals during the AHB dataphase.There are three signals from target to controller, hready, hresp0, and hresp2. If any of the controller signals is different from the target signals during dataphase, the error will be reported in the Error Status Register.

Sample Reference Manual

Chapter 9
Crossbar Switch (XBAR)

9.1 Chip-specific XBAR information

This chip has one instance of the XBAR module.

9.1.1 XBAR controller and target assignments

The following table lists the XBAR physical port numbers and logical IDs for all controller ports on this SoC.

- Each port number matches the default priority assigned to the corresponding physical controller port. This default priority equals the reset value of the priority field for each controller port in the PRSn registers.
- A priority value of 0 is the highest priority. There is no "disabled" value for the priority.
- A Nexus_3 module and core data bus share the same physical controller port for each core.

The logical controller ID corresponds to the logical address provided by the controller module and is unique for each module. The logical controller IDs are used by the bus controllers connected to the XBAR. The Nexus controller is identified by setting the MSB in the 4-bit field that supplies the controller ID number.

Table 9-1.XBAR controller ports and logical controller IDs

Module	Physical controller port	Logical controller ID	Comment
Core0 instruction	0	0	
Core0 data	1	0	
Nexus_3_0		8	Nexus_3_0 arbitrates with Core0 data for XBAR port 1
Core1 instruction	2	1	
Core1 data	3	1	
Nexus_3_1		9	Nexus_3_1 arbitrates with Core1 data for XBAR port 3

Table continues on the next page...

Sample Reference Manual

Figure 3. Example of chip-specific information that refers to a different chapter

1.4 Register descriptions

Module chapters present register information in the following:

- Memory maps, which contain:
  - An offset from the module's base address
  - The mnemonic and name of each register
  - The width of each register (in bits)
  - The reset value of each register
- Register figures
- Field-description tables
- Associated text

The following figure shows register figure conventions used throughout the manual.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

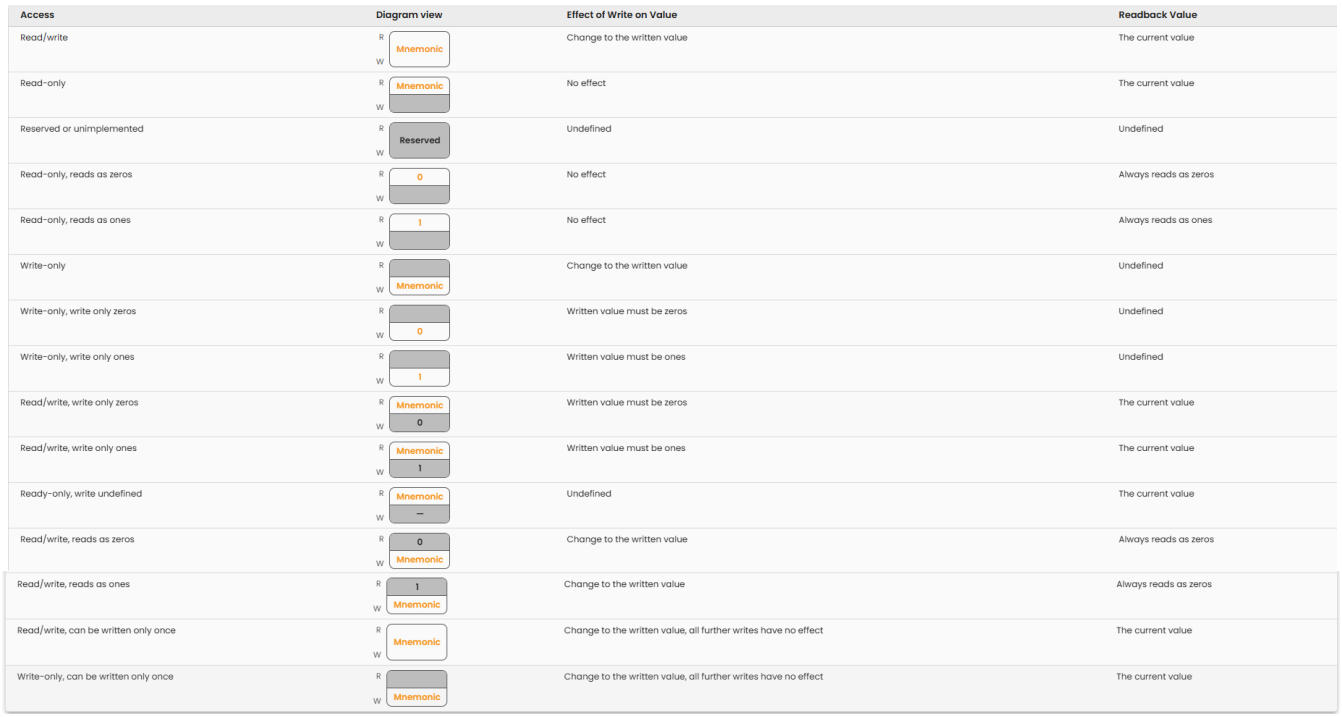


Figure 4. Register access types

NOTE

Reset values of reserved locations documented in this manual are subject to change and must not be used for diagnostic purposes.

Table 1. Modified write values

Modified write values	Description
Write one to clear	Writing a 1 clears the corresponding bit in the field to zero. Writing 0 has no effect.
Write one to set	Writing a 1 sets the corresponding bit in the field to one. Writing 0 has no effect.
Write one to toggle	Writing a 1 toggles the corresponding bit in the field. Writing 0 has no effect.
Write zero to clear	Writing a 0 clears the corresponding bit in the field to zero. Writing 1 has no effect.
Write zero to set	Writing a 0 sets to one the corresponding bit in the field to one. Writing 1 has no effect.
Write zero to toggle	Writing a 0 toggles the corresponding bit in the field. Writing 1 has no effect.
Write always clears	Writing any value clears all bits in the field to zeros.
Write always sets	Writing any value sets all bits in the field to ones.
Modification in an unspecified way	Writing any value modifies all bits in the field in an unspecified way.

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table 1. Modified write values (continued)

Modified write values	Description
Write the value as is	The written value is stored in the field as-is. The most common behavior for control and data registers.

Table 2. Read action

Read action	Description
Read to clear	The field is cleared to zeros after the read operation.
Read to set	The field is set to ones after the read operation.
Read to modify	The field is modified in an unspecified way after the read operation.
Read to modify external	The field, and possibly fields in other registers, are modified in an unspecified way after the read operation.
Read without any side effect	The field is not modified after the read operation. The most common behavior for control and data registers.

1.5 Conventions

1.5.1 Notes and cautions

Specific information is provided as part of notes and cautions throughout this manual.

NOTE

Emphasizes information that deserves extra attention.

CAUTION

Informs you of situations that could lead to highly undesirable outcomes—such as damage to the chip or irreversible malfunction.

1.5.2 Numbering systems

The following suffixes identify different numbering systems:

Table 3. Numbering systems

This suffix	Identifies a
b	Binary number. For example, the binary equivalent of the number 5 is mentioned as 101b. In some cases, *0b* is prefixed to binary numbers.
d	Decimal number. Decimal numbers are followed by this suffix only when there is a possibility of confusion. In general, decimal numbers are used without a suffix.
h	Hexadecimal number. For example, the hexadecimal equivalent of the number 60 is mentioned as 3Ch. In some cases, *0x* is prefixed to hexadecimal numbers.

1.5.3 Typographic notation

The following typographic notations are used throughout this document:

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table 4. Typographic notation

Example	Description
x and other italicized text	The italicized, lowercase *x* is used as a placeholder for replaceable numbers. In general, italicized text is used for titles of publications and for emphasis. Additionally, italics could be used for metasymbols in syntax descriptions. Plain lowercase letters are used as placeholders for single letters and numbers.
`code font`	Fixed-width font (such as Courier) used for code. It is used for a letter, word, or phrase that you want the user to type. For example, "Type `Read` and press Enter." This type of font is also used for instruction mnemonics, directives, symbols, subcommands, parameters, operators, computer-language elements, code listings, commands that appear in running text, and for sample code. Instruction mnemonics and directives in text and tables are mentioned in all caps; for example, BSR.
SR[SCM]	A mnemonic in square brackets represents the name of a register field. This example refers to the Scaling Mode (SCM) field in the Status Register (SR).
REVNO[6:4], XAD[7:0]	Numbers in brackets that are separated by a colon represent either: • A subset of a register's named field For example, REVNO[6:4] refers to bits 6-4 that are part of the COREREV field occupying bits 6-0 of the REVNO register. • A continuous range of individual signals of a bus For example, XAD[7:0] refers to signals 7-0 of the XAD bus.
MOD.REG	A period separates the elements of a hierarchy: subsystem.module.register. For example: • SWT.TO means that the TO register is located in the SWT module. • SMU.XRDC.CR means that the CR register is located in the XRDC module within the SMU subsystem.

1.5.4 Special terms

The following terms have special meanings.

Table 5. Special terms

Term	Meaning
Asserted	Refers to the state of a signal as follows: • An active-high signal is asserted when high (1). • An active-low signal is asserted when low (0).
Deasserted	Refers to the state of a signal as follows: • An active-high signal is deasserted when low (0). • An active-low signal is deasserted when high (1). In some cases, deasserted signals are described as *negated*.
Reserved	Refers to memory space, register, field, or programming setting. Writes to a reserved location can result in unpredictable functionality or behavior. You must:

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table 5. Special terms (continued)

Term	Meaning
	• Before writing to a location which contain reserved bits user must make sure the write operation will write the reserved bit with value specified as the reset value in NXP reference manual. • Consider undefined locations in memory to be reserved as reset value 0 shall be assumed. You might get a BERR(transfer error) response on access to undefined locations in memory. • If user reads data from memory area containing reserved bit, the value of reserved bits should be ignored and not used for any functional purposes. **NOTE** BootROM could modify the reserved bit values after reset. Please refer to the BootROM settings attachment.
Write 1 to clear (w1c)	Refers to the access type of a register field that is used to clear the field by writing the value 1 to it.
Undefined (u)	Refers to undefined reset values

1.6 Editorial changes

Each new release of this document includes editorial improvements such as:

- Spelling
- Grammar
- Punctuation
- Voice
- Tense
- Capitalization
- Formatting
- Presentation
- Navigation

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Chapter 2
Introduction

2.1 Overview

This product family extends NXP's essential micro controller (MCU) portfolio by offering a 72 MHz Arm® Cortex®-M23 core, offering a balance of cost-efficiency, low power consumption, and performance through comprehensive set of peripherals, including a 16-bit analog-to-digital converter (ADC) and advanced communication interfaces for high-speed data exchange, making it suitable for a wide range of embedded applications.

2.2 Target applications

This device has following target applications:

- Home Appliances
- Remote Controls
- Electronic Toys
- Industrial Sensors
- Simple Motor Control
- LED Lighting Control
- Basic Internet of Things (IoT) Nodes
- Human-Machine Interface (HMI)
- Security Systems
- Medical Devices
- Power Tools

2.3 Block diagram

The following figure shows a top-level organization of the modules within the chip organized by functional category.

All information provided in this document is subject to legal disclaimers.
© 2026 NXP B.V. All rights reserved.

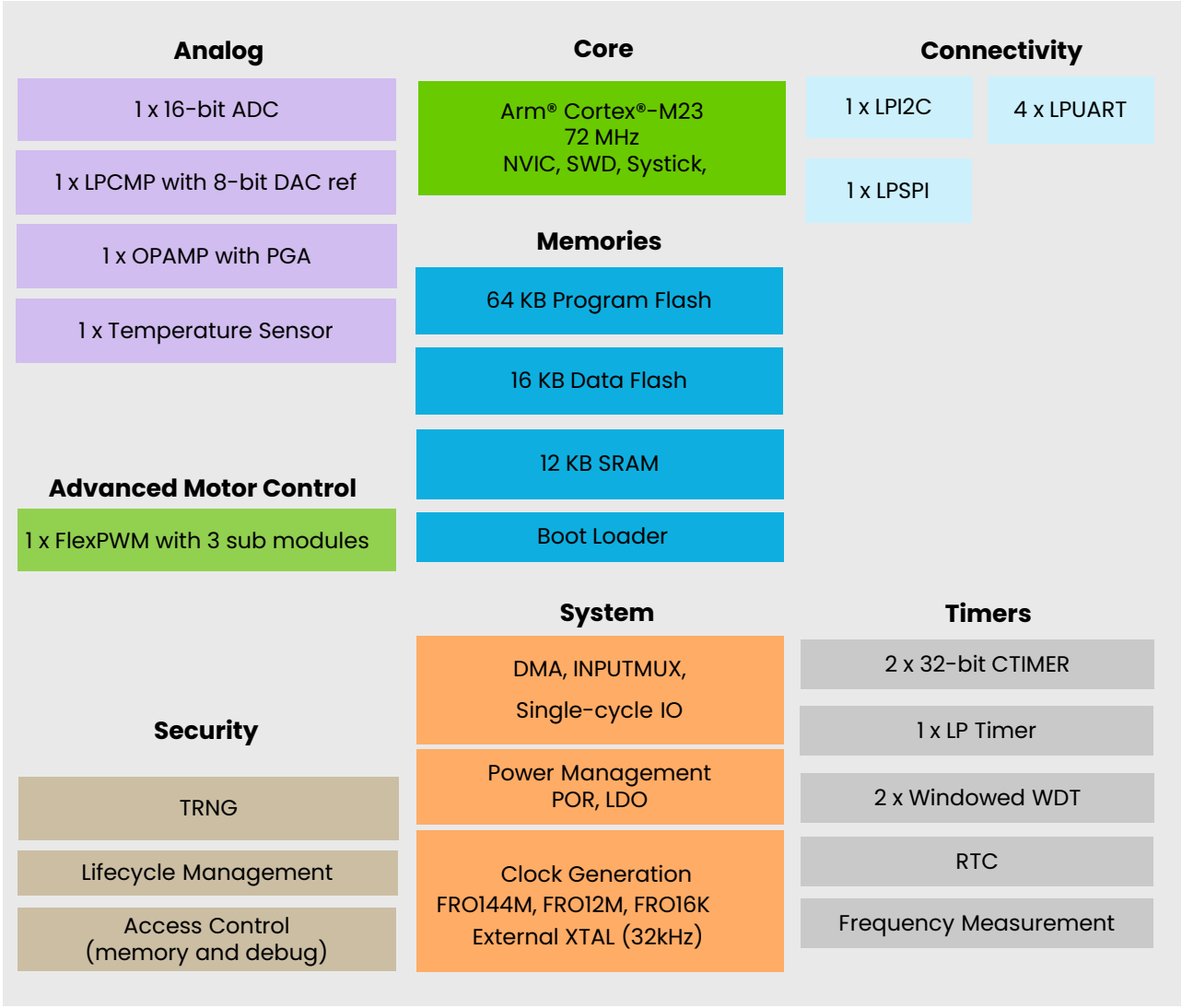


Figure 5. Features block diagram

2.4 Features

This product family offers the following features:

- Core
  - Arm® 32-bit Cortex®-M23 CPU, with single cycle multiplier and 21-cycle fast divider, without Memory Protection Unit (MPU) or Trust Zone® support
  - Nested Vectored Interrupt Controller
  - Clock gating for core processor
  - Multilayer AHB Bus Matrix
- Memories
  - Up to 64 KB Flash array implemented as single array. The smallest programming phrase should be 16-bytes.
    - 128-bit + 9-bit ECC, 8 KB sector size
    - Up to 16 KB of data flash, 2 sectors with 8 KB per sector

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

    - 32 KB IFR0 + 8 KB IFR1 for Implicit-protected Flash Region (IFR)
  - Flash Memory Controller
    - Line buffer
    - Prefetch buffer
    - Single wait state at 72 MHz (SD mode)
  - Memory Block Checker (MBC)
    - Protect flash read, write and execute permission
    - Main array granularity is 8 KB
    - IFR0 granularity is 2 KB
    - IFR1 granularity is 2 KB
  - RAM
    - 12 KB SRAM (8 KB with ECC)
- Timers
  - 32-bit standard general-purpose asynchronous timer/counter (CTIMER), which supports up to four capture inputs and four compare outputs, PWM mode, and external count input. Specific timer events can be selected to generate DMA requests.
  - Real time clock (RTC)
  - Low-power timer (LPTMR)
  - Frequency measurement timer
  - Windowed watchdog timer (WWDT)
- Advanced Motor Control
  - Single FlexPWM
    - 3 sub-modules per each FlexPWM, providing 6 complementary outputs of PWM (no Nanoedge module)
    - 4 fault inputs are supported
- Communication Interfaces for Connectivity
  - 1x LPI2C, 1x LPSPI, 4x LPUART
- Analog
  - 1x 16-bit ADC
    - Up to 2.4 Msps in 16-bit mode, and 3 Msps in 12-bit mode
    - Up to 17 ADC Input channels (depending on the package)
    - One integrated temperature sensor per ADC
  - 1x High-speed Comparators with 8 input pins and 8-bit DAC as internal reference
    - The analog functionality of the LPCMP remain operational down to Deep Power Down (DPD) mode
  - 1x OPAMP with PGA
- Operating characteristics
  - Operating voltage: 1.71 to 3.6 V
  - Temperature range: - 40 to 125 °C
- Debug
  - Minimal debug – 2 breakpoint and 2 watch point, no ETM

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

  - Debug mailbox allows the communication between CPU and tools to ensure data consistency during power up.
  - SWD
- Packages
  - 48-pin LQFP (LQFP48)
  - 48-pin H-PQFN (H-PQFN48)
  - 32-pin H-PQFN (H-PQFN32)
  - 24-pin H-PQFN (H-PQFN24)
  - 16-pin H-PQFN (H-PQFN16)

See the chip data sheet for details on packages.

2.5 Functional overview

The following table shows the chip modules organized by functional category.

Table 6. Module functional categories

Module category	Description
Core	• The Arm Cortex-M23 is a member of the Cortex-M Series of processors targeting micro controller cores focused on cost sensitive, deterministic, and interrupt driven environments. The Cortex-M23 processor is based on the Armv8-M Architecture and ThumbR-2 ISA and is upward compatible with the Cortex-M7, M4, M3, M33, and M0/M0+.
System	• Debug Mailbox (DBGMB) • AHB Cross-Bar Switch (AXBS) Lite • Enhanced DMA Controller (4-channel eDMA3) • Wakeup Unit (WUU) • Input Multiplexing (INPUTMUX) • Error Injection Module (EIM) • Error Reporting Module (ERM) • System Power Control (SPC) • System Controller (SYSCON) • RMC (part of CMC)
Memory	• Flash Memory Controller (FMC) • Flash Memory Module (FMU) • Static Random Access Memory (SRAM)
Clock	• VBAT Wrapper — Free Running Oscillator - 16 kHz (FRO16K) — Crystal Oscillator - 32 kHz (OSC32K) • System Clock Generator (SCG)

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table 6. Module functional categories (continued)

Module category	Description
	— Free Running Oscillator - 144 MHz (FRO144M) — Free Running Oscillator - 12 MHz (FRO12M)
Security	• Cyclic Redundancy Check (CRC) • Memory Block Checker (MBC) • True Random Generator (TRNG)
Timer	• Low-power Timer (LPTMR) • Standard Counter/Timer (CTIMER) • Windowed Watchdog Timer (WWDT) • Frequency Measurement (FREQME) • Real-time counter (RTC)
Communication	• Low-Power UART (LPUART) • Low-Power Serial Peripheral Interface (LPSPI) • Low-Power I2C (LPI2C)
Human Machine Interface (HMI)	• General Purpose Input/Output (GPIO) • Port Control (PORT)
Analog	• 16-bit analog-to-digital converter (ADC) with 3 Msps conversion rate in 12-bit mode, and 2.4 Msps in 16-bit mode • Operational Amplifier (OPAMP) • Low Power Comparator (LPCMP)

2.5.1 Core

The following core modules are available on this chip.

Table 7. Core modules

Module	Description
CPU	Is an Arm Cortex-M23 processor that runs at a frequency of up to 72 MHz.
Nested Vectored Interrupt Controller (NVIC)	The Armv8M exception model and Nested-Vectored Interrupt Controller (NVIC) implement a relocatable vector table supporting external interrupts, a single non-maskable interrupt (NMI), and priority levels.
System Tick Timer (SysTick)	See the Armv8M Architecture Reference Manual for more information about this system timer.

2.5.2 Clock

The following clock modules are available on this chip.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table 8. Clock sources

Module	Description
Free Running Oscillator - 16 kHz (FRO16K)	Is an ultra-low power internal 16.384 kHz clock source. It is functional down to Deep Power Down (DPD) mode. The FRO16K is trimmed to ±6% accuracy over the entire voltage and temperature range.
System Clock Generator (SCG)	Provides the user with access to the configuration control registers for the system level clock sources.
Crystal Oscillator - 32 kHz (OSC32K)	Generates, in conjunction with an external crystal or resonator, a reference clock for the chip.
Free Running Oscillator - 144 MHz (FRO144M)	The free running oscillator can generate 144 MHz for use by the chip.
Free Running Oscillator - 12 MHz (FRO12M)	Is an internal clock source that generates a 12 MHz frequency for use by the chip.

2.5.3 Communication

The following communication modules are available on this chip.

Table 9. Communication modules

Module	Description
Low Power Inter-Integrated Circuit (LPI2C)	Allows communication between a number of devices. Also supports the System Management Bus (SMBus) Specification, version 2.
Low Power Serial Peripheral Interface (LPSPI)	Synchronous serial bus for communication to an external device.
Low Power Universal Asynchronous Receive/ Transmit (LPUART)	Asynchronous serial bus communication interface with programmable 8- or 9-bit data format.

2.5.4 Debug

The following debug modules are available on this chip.

Table 10. Debug modules

Module	Description
Data Watchpoint and Trace (DWT)	Is a generic name for modules that allow debug access of the Cortex-M23. The DWT comprises of the Debug Watchpoint and Trace (DWT) module and the Flash Patch and Breakpoint (FPB) unit.
Debug Access Port (DAP)	Enables real-time access to the chip registers from an external debugger without halting the processor cores.
Serial Wire Debug (SWD)	Is a serial communication interface used for debugging devices with multiple cores while only requiring a single external interface.

2.5.5 Memory

The following memory modules are available on this chip.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table 11. Memory modules

Module	Description
Flash Memory Controller (FMC)	Is a programmable flash memory — non-volatile flash memory that can store executable program code or data.
Flash Management Unit (FMU)	Manages the interface between the chip and the on-chip flash memory.
Static Random Access Memory (SRAM)	Internal system SRAM memory. Each individual block of SRAM can be configured to be retained in low-power modes.

2.5.6 System

The following system modules are available on this chip.

Table 12. System modules

Module	Description
Peripheral Bridge (PBRIDGE)	Converts an AMBA AHB interface to a peripheral interface that allows the Peripheral Bridge (PBRIDGE) controller to interface to multiple peripherals. This device has two peripheral bridges.
System Controller (SYSCON)	Provides controls and configurations of the system and peripherals for the multiple functions.
Security Controller (SECCON)	Provides control and configuration of security related functions. The information is included in the SYSCON
Core Mode Controller (CMC)	The CMC provides control and protection on entry and exit to each power mode, control for the System Power Controller (SPC), and reset entry and exit for the complete device.
Enhanced Direct Memory Access (eDMA)	Performs source/destination address calculations and data-movement operations. Capable of performing complex data transfers with minimal intervention from a host processor.
Wake-up Unit (WUU)	Allows selection of external pins and internal modules as interrupt wake-up sources from Deep Sleep and Power Down modes.
Peripheral Input Multiplexing (INPUTMUX)	Allows the trigger output of one peripheral to be connected to the trigger input of a second peripheral.
Error Injection Module (EIM)	Provides a method for diagnostic coverage of internal memories (for example, system RAM, cache RAMs, and peripheral memories).
Error Reporting Module (ERM)	Provides information and optional interrupt notification on memory error events associated with Error correction code (ECC) and parity.
System Power Controller (SPC)	Provides control over the operation and configuration of the system power generation modules to optimize power consumption for the level of functionality needed.

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table 12. System modules (continued)

Module	Description
Low Drop Out Regulator - Core (LDO_CORE)	A voltage regulator for generating the core voltage for the chip.
RAM_RET_LDO	A voltage regulator for generating the voltage for SRAM retention.

2.5.7 Security

The following security modules are available on this chip.

Table 13. Security modules

Module	Description
Cyclic Redundancy Check (CRC)	Provides error detection for all single, double, and many multi-bit errors.
Memory Block Checker (MBC)	Provides domain-based access control for all system bus references targeted to on-chip internal memories and slave peripherals. It's a part of TRDC module.
True Random Number Generator (TRNG)	It s a hardware accelerator module that generates 256-bit entropy as needed by an entropy-consuming module or by other post-processing functions.

2.5.8 Timer

The following timer modules are available on this chip.

Table 14. Timer modules

Module	Description
Flex Pulse Width Modulator (FlexPWM)	Generates various switching patterns, including highly sophisticated waveforms. Controls different Switched Mode Power Supplies (SMPS) topologies.
Low-power Timer (LPTMR)	Is a 16-bit timer or pulse counter with compare feature.
Standard Counter/Timer (CTIMER)	Each Counter/timer is designed to count cycles of the CTIMER function clock.
Windowed Watchdog Timer (WWDT)	Helps reset or interrupt an erroneous micro controller within a programmable time.
Frequency Measurement (FREQME)	Provides high-accuracy frequency measurement function for on-chip and off-chip clocks.

2.5.9 Human Machine Interface (HMI)

The following HMI modules are available on this chip.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table 15. HMI modules

Module	Description
General Purpose Input/Output (GPIO)	All GPIO pins support interrupt and DMA request generation.
Port Control (PORT)	Provides support for pad control functions.

2.5.10 Analog

The following analog modules are available on this chip.

Table 16. Analog modules

Module	Description
Analog-to-Digital Converter (ADC)	Successive approximation ADC designed for operation within an integrated micro controller system-on-chip.
Low Power Comparator (LPCMP)	Provides a circuit for comparing two analog input voltages.
Operational Amplifier (OPAMP)	The OPAMP is a pure voltage amplifier with a built-in programmable gain stage.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Chapter 3
Core Overview

3.1 Introduction

This section covers the core modules included in this chip.

3.2 Cortex-M23 core

This chip has one Arm® Cortex®-M23 processor core. This core is a low gate count, two-stage, and highly energy efficient processor. It is for microcontroller and deeply embedded applications that require an area optimized, low-power processor for use in environments where security is an important consideration.

The core on this chip supports below features:

- Fetch instructions using 32-bit AHB-Lite accesses whenever possible.
- Little-endian (least significant byte at lowest byte address location)
- Single cycle 32×32 multiply instruction
- 21-cycle divider
- SYSTICK timer
- Two hardware breakpoints and two hardware watchpoints for debug

The interfaces in the processor for external access include:

- External AMBA® AHB5 interface

  The AHB interface is also named as system bus. CPU fetch instruction and data from this bus.
- Debug Access Port (DAP)
- Single-cycle I/O port

3.2.1 System bus access

All System bus accesses are routed to the target address in destination memories through multilayer AHB matrix slave port.

3.2.2 Access control

The SRAM supports write permission access control with the control registers in SECCON.

The MBC provides read, write and execute permission control to flash.

Debug permission control is supported through life cycle and control registers are in SECCON.

NOTE

MPU and Trustzone are not supported.

3.2.3 I/O port

Arm® Cortex®-M23 core on this chip provides I/O ports which provide the separated interface to access GPIO with single cycle. There is an arbiter between I/O port, IPS bridge and GPIO. With the arbiter, the GPIO can be accessed from either I/O port or bus fabric. Software must disable GPIO auto clock gating when using I/O port to access GPIO.

The figure below shows the I/O port.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

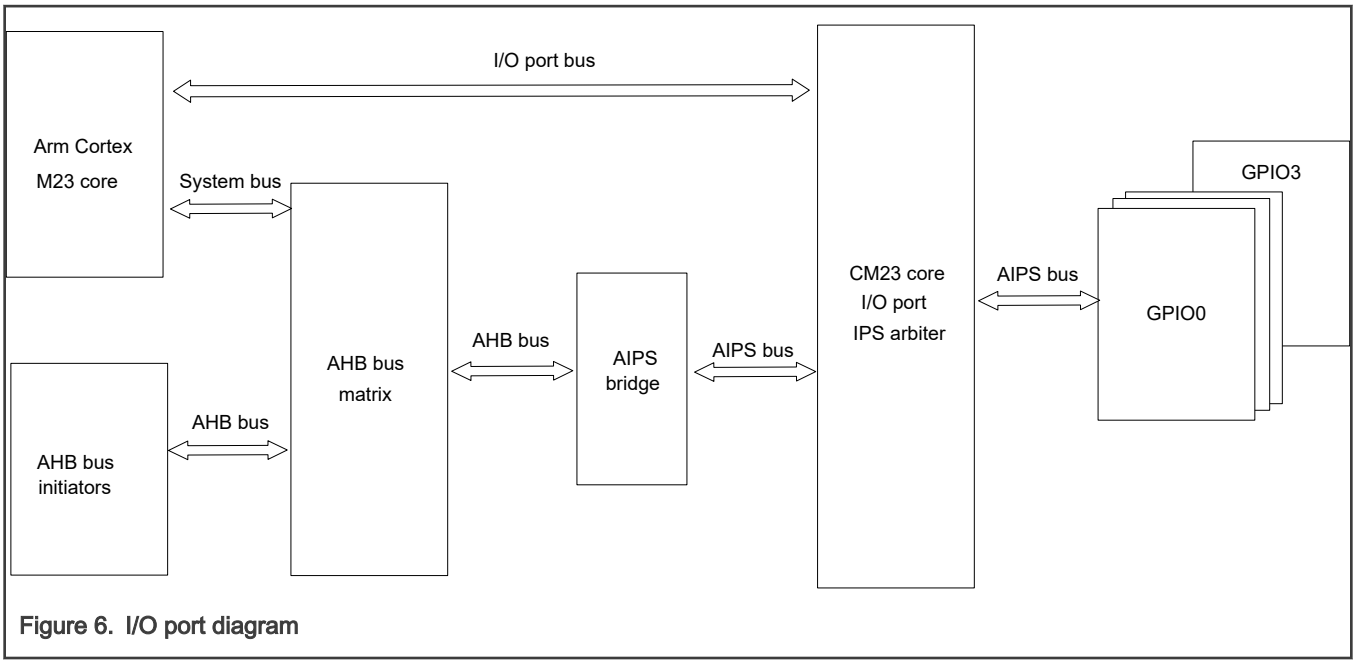


Figure 6. I/O port diagram

GPIO address is mapped to different space when accessing GPIO from different bus as shown in the following table.

Table 17. GPIO base address

GPIO instances	AHB access base address	I/O port access base address
GPIO0	400F_3000h	F800_1000h
GPIO1	400F_4000h	F800_2000h
GPIO2	400F_5000h	F800_3000h
GPIO3	400F_6000h	F800_4000h

3.3 Nested Vectored Interrupt Controller (NVIC)

3.3.1 Interrupt priority levels

This device supports 8 priority levels for interrupts. Therefore, in the NVIC each source in the IPR registers contains 3 bits. For example, IPR0 is shown below:

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	IRQ3			0	0	0	0	0	IRQ2			0	0	0	0	0	IRQ1			0	0	0	0	0	IRQ0			0	0	0	0	0
W																																

3.3.2 Non-Maskable Interrupt (NMI) configuration

The Non-Maskable Interrupt (NMI) enable field and source selection field are configured by NMI Source Select (NMISRC). See Table 19 for the sources available.

3.3.3 Interrupt channel assignments

The interrupt source assignments are defined in the following table.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

- Vector number — the value stored on the stack when an interrupt is serviced.
- IRQ number — non-core interrupt source count, which is the vector number minus 16.
- NVIC non-IPR register number — NVIC's ISER, ICER, ISPR, ICPR, and IABR register number used for this IRQ. The equation to calculate this value is: IRQ number/32.
- NVIC IPR register number — NVIC's IPR register number used for this IRQ. The equation to calculate this value is: IRQ number/4.

The IRQ number is used within Arm's NVIC documentation.

See the following table for the NVIC interrupt assignments.

Table 19. NVIC configuration

Cortex-M23 core vector offset address	Vector number	IRQ number	NVIC non-IPR register number	NVIC IPR register number	Module instance	Interrupt type
0	0	-	-	-	Cortex-M23	Initial stack pointer
4	1	-	-	-	Cortex-M23	Initial program counter
8	2	-	-	-	Cortex-M23	Non-maskable interrupt (NMI)
C	3	-	-	-	Cortex-M23	Hard fault
10	4	-	-	-	Reserved	
14	5	-	-	-	Reserved	
18	6	-	-	-	Reserved	
1C	7	-	-	-	Reserved	
20	8	-	-	-	Reserved	
24	9	-	-	-	Reserved	
28	10	-	-	-	Reserved	
2C	11	-	-	-	Cortex-M23	Supervisor call (SVCall)
30	12	-	-	-	Reserved	
34	13	-	-	-	Reserved	
38	14	-	-	-	Cortex-M23	Pendable request for system service (Penable SrvReq)
3C	15	-	-	-	Cortex-M23	System tick timer
40	16	0	0	0	Reserved0	OR IRQ[1] to IRQ[90]
44	17	1	0	0	CMC0	Core mode controller interrupt
48	18	2	0	0	DMA0	DMA3_0_CH0 error or transfer complete
4C	19	3	0	0	DMA0	DMA3_0_CH1 error or transfer complete

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table 19. NVIC configuration (continued)

Cortex-M23 core vector offset address	Vector number	IRQ number	NVIC non-IPR register number	NVIC IPR register number	Module instance	Interrupt type
50	20	4	0	1	DMA0	DMA3_0_CH2 error or transfer complete
54	21	5	0	1	DMA0	DMA3_0_CH3 error or transfer complete
58	22	6	0	1	Reserved	
5C	23	7	0	1	Reserved	
60	24	8	0	2	Reserved	
64	25	9	0	2	Reserved	
68	26	10	0	2	ERM0	ERM single-bit error interrupt
6C	27	11	0	2	ERM0	ERM multi-bit error interrupt
70	28	12	0	3	FMU0	Flash management unit interrupt
74	29	13	0	3	Reserved	
78	30	14	0	3	MBC0	MBC secure violation interrupt
7C	31	15	0	3	SCG0	System clock generator interrupt
80	32	16	0	4	SPC0	System power controller interrupt
84	33	17	0	4	TRNG0	TRNG Interrupt or TRNG Error
88	34	18	0	4	WUU0	Wake up unit interrupt
8C	35	19	0	4	Reserved	
90	36	20	0	5	Reserved	
94	37	21	0	5	Reserved	
98	38	22	0	5	Reserved	
9C	39	23	0	5	Reserved	
A0	40	24	0	6	Reserved	
A4	41	25	0	6	Reserved	
A8	42	26	0	6	LPI2C0	Low-power inter integrated circuit interrupt
AC	43	27	0	6	Reserved	
B0	44	28	0	7	LPSPI0	Low-power serial peripheral interface interrupt
B4	45	29	0	7	Reserved	
B8	46	30	0	7	Reserved	

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table 19. NVIC configuration (continued)

Cortex-M23 core vector offset address	Vector number	IRQ number	NVIC non-IPR register number	NVIC IPR register number	Module instance	Interrupt type
BC	47	31	0	7	LPUART0	Low-power universal asynchronous receive/transmit interrupt
C0	48	32	1	8	LPUART1	Low-power universal asynchronous receive/transmit interrupt
C4	49	33	1	8	LPUART2	Low-power universal asynchronous receive/transmit interrupt
C8	50	34	1	8	LPUART3	Low-power universal asynchronous receive/transmit interrupt
CC	51	35	1	8	Reserved	
D0	52	36	1	9	Reserved	
D4	53	37	1	9	Reserved	
D8	54	38	1	9	Reserved	
DC	55	39	1	9	CTIMER0	Standard counter/timer 0 interrupt
E0	56	40	1	10	CTIMER1	Standard counter/timer 1 interrupt
E4	57	41	1	10	Reserved	
E8	58	42	1	10	Reserved	
EC	59	43	1	10	Reserved	
F0	60	44	1	11	FlexPWM0	FlexPWM0_reload_error interrupt
F4	61	45	1	11	FlexPWM0	FlexPWM0_fault interrupt
F8	62	46	1	11	FlexPWM0	FlexPWM0 sub-module 0 capture/ compare/reload interrupt
FC	63	47	1	11	FlexPWM0	FlexPWM0 sub-module 1 capture/ compare/reload interrupt
100	64	48	1	12	FlexPWM0	FlexPWM0 sub-module 2 capture/ compare/reload interrupt
104	65	49	1	12	Reserved	
108	66	50	1	12	Reserved	
10C	67	51	1	12	Reserved	
110	68	52	1	13	Reserved	
114	69	53	1	13	Reserved	
118	70	54	1	13	FREQME0	Frequency measurement interrupt
11C	71	55	1	13	LPTMR0	Low power timer 0 interrupt

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table 19. NVIC configuration (continued)

Cortex-M23 core vector offset address	Vector number	IRQ number	NVIC non-IPR register number	NVIC IPR register number	Module instance	Interrupt type
120	72	56	1	14	Reserved	
124	73	57	1	14	VBAT0	VBAT OSC32K ready interrupt
128	74	58	1	14	RTC0	RTC alarm interrupt
12C	75	59	1	14	RTC0	RTC 1 Hz interrupt
130	76	60	1	15	WWDT0	Windowed watchdog timer 0 interrupt
134	77	61	1	15	WWDT1	Windowed watchdog timer 1 interrupt
138	78	62	1	15	ADC0	Analog-to-digital converter interrupt
13C	79	63	1	15	Reserved	
140	80	64	2	16	CMP0	Comparator interrupt
144	81	65	2	16	Reserved	
148	82	66	2	16	Reserved	
14C	83	67	2	16	Reserved	
150	84	68	2	17	Reserved	
154	85	69	2	17	Reserved	
158	86	70	2	17	Reserved	
15C	87	71	2	17	GPIO0	General purpose input/output interrupt 0
160	88	72	2	18	GPIO1	General purpose input/output interrupt 0
164	89	73	2	18	GPIO2	General purpose input/output interrupt 0
168	90	74	2	18	GPIO3	General purpose input/output interrupt 0

3.4 System memory map

The following table provides the chip's high-level memory map.

Table 20. System memory map

Start address (hex)	End address (hex)	Size (KB)	Description	Access
Code bus memory				
0000_0000	0000_FFFF	64	Program flash (target port 0)	All initiators
0001_0000	0001_BFFF	48	NXP bootloader	All initiators

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table 20. System memory map (continued)

Start address (hex)	End address (hex)	Size (KB)	Description	Access
0001_C000	0001_FFFF	16	User data flash	All initiators
0002_0000	00FF_FFFF	16256	Reserved	
0100_0000	0100_1FFF	8	IFR 0 sector 0 (target port 0, NXP bootloader)	All initiators
0100_2000	0100_3FFF	8	IFR 0 sector 1 (target port 0, CMPA)	All initiators
0100_4000	0100_7FFF	16	IFR 0 sector 2/3 (target port 0, user usage)	All initiators
0100_8000	010F_FFFF	992	Reserved	
0110_0000	0110_1FFF	8	IFR 1 (target port 0)	All initiators
0110_2000	02FF_FFFF	31736	Reserved	
0300_0000	0300_5FFF	24	Reserved	
0300_6000	03FF_FFFF	16360	Reserved	
0400_0000	0400_1FFF	8	Reserved	All initiators
0400_2000	0400_2FFF	4	Reserved	All initiators
0400_3000	1FFF_FFFF	458740	Reserved	
System RAM				
2000_0000	2000_1FFF	8	RAM A0 (target port 1)[1]	All initiators
2000_2000_	2000_2FFF	4	RAM A1 (target port 1)[2]	All initiators
2000_3000	2000_3FFF	4	Reserved	All initiators
2000_4000	2000_7FFF	16	Reserved	All initiators
2000_8000	2000_FFFF	32	Reserved	All initiators
2001_0000	2001_7FFF	32	Reserved	All initiators
2001_8000	2001_FFFF	32	Reserved	All initiators
2002_0000	2002_7FFF	32	Reserved	All initiators
2002_8000	2002_FFFF	32	Reserved	All initiators
2003_0000	2003_7FFF	32	Reserved	All initiators
2003_8000	2003_BFFF	16	Reserved	All initiators
2003_C000	2003_DFFF	8	Reserved	All initiators
2003_E000	3FFF_FFFF	524040	Reserved	
Peripheral groups				
4000_0000	4001_FFFF	128	Peripheral bridge 0 (target port 2)	All initiators
4002_0000	4007_FFFF	384	Reserved	

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table 20. System memory map (continued)

Start address (hex)	End address (hex)	Size (KB)	Description	Access
4008_0000	400F_FFFF	512	Peripheral bridge 1 (target port 2)	All initiators
4010_0000	4010_8FFF	36	Fast peripherals (target port 2)	All initiators
4010_9000	4FFF_FFFF	261084	Reserved	
5000_0000	5FFF_FFFF	262144	Reserved	
6000_0000	6FFF_FFFF	262144	Reserved	
7000_0000	7FFF_FFFF	262144	Reserved	
8000_0000	8FFF_FFFF	262144	Reserved	
9000_0000	9FFF_FFFF	262144	Reserved	
A000_0000	AFFF_FFFF	262144	Reserved	
B000_0000	BFFF_FFFF	262144	Reserved	
C000_0000	CFFF_FFFF	262144	Reserved	
D000_0000	DFFF_FFFF	262144	Reserved	
E000_0000	E003_FFFF	256	Private peripheral bus (internal)	Core only
E004_0000	E00F_FFFF	768	Private peripheral bus (external) (includes NVIC and SYSTICK timer)	Core only
E010_0000	FFFF_FFFF	523264	Reserved	

1. For the chip with 6 KB RAM, the end address is 2000_17FF.
2. Reserved for the chip with 6 KB RAM.

3.5 Peripheral bridge

The Peripheral Bridge (PBRIDGE) is the portion of the bus fabric that connects the peripherals to the processor elements. Each peripheral has a base address where the processor elements can access them.

3.5.1 Peripheral bridge (PBRIDGE0-1) memory maps

See the following table for PBRIDGE0, PBRIDGE1, and Fast peripherals memory map on this chip.

Table 21. Peripheral memory map

Start address (hex)	End address (hex)	Size (KB)	Peripheral instance
Peripheral bridge 0			
4000_0000	4000_0FFF	4	Reserved
4000_1000	4000_1FFF	4	INPUTMUX0
4000_2000	4000_3FFF	8	Reserved
4000_4000	4000_4FFF	4	CTIMER0
4000_5000	4000_5FFF	4	CTIMER1

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table 21. Peripheral memory map (continued)

Start address (hex)	End address (hex)	Size (KB)	Peripheral instance
4000_6000	4000_8FFF	12	Reserved
4000_9000	4000_9FFF	4	FREQME0
4000_A000	4000_BFFF	8	Reserved
4000_C000	4000_CFFF	4	WWDT0
4000_D000	4000_DFFF	4	WWDT1
4000_E000	4001_FFFF	72	Reserved
Peripheral bridge 1			
4008_0000	4008_0FFF	4	eDMA_0_MP
4008_1000	4008_1FFF	4	eDMA_0_TCD0
4008_2000	4008_2FFF	4	eDMA_0_TCD1
4008_3000	4008_3FFF	4	eDMA_0_TCD2
4008_4000	4008_4FFF	4	eDMA_0_TCD3
4008_5000	4008_9FFF	20	Reserved
4008_A000	4008_AFFF	4	CRC0
4008_B000	4008_BFFF	4	CMC
4008_C000	4008_CFFF	4	EIM0
4008_D000	4008_DFFF	4	ERM0
4008_E000	4008_EFFF	4	MBC0
4008_F000	4008_FFFF	4	SCG0
4009_0000	4009_0FFF	4	SPC0
4009_1000	4009_1FFF	4	SYSCON
4009_2000	4009_2FFF	4	WUU0
4009_3000	4009_3FFF	4	VBAT0
4009_4000	4009_4FFF	4	FMC0
4009_5000	4009_5FFF	4	FMU0
4009_6000	4009_6FFF	4	FMU0_TST
4009_7000	4009_9FFF	12	Reserved
4009_A000	4009_AFFF	4	LPI2C0
4009_B000	4009_BFFF	4	Reserved
4009_C000	4009_CFFF	4	LPSPI0
4009_D000	4009_EFFF	4	Reserved
4009_F000	4009_FFFF	4	LPUART0

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table 21. Peripheral memory map (continued)

Start address (hex)	End address (hex)	Size (KB)	Peripheral instance
400A_0000	400A_0FFF	4	LPUART1
400A_1000	400A_1FFF	4	LPUART2
400A_2000	400A_2FFF	4	LPUART3
400A_3000	400A_8FFF	24	Reserved
400A_9000	400A_9FFF	4	FlexPWM0
400A_A000	400A_AFFF	4	Reserved
400A_B000	400A_BFFF	4	LPTMR0
400A_C000	400A_EFFF	12	Reserved
400A_F000	400A_FFFF	4	ADC0
400B_0000	400B_0FFF	4	Reserved
400B_1000	400B_1FFF	4	LPCMP0
400B_2000	400B_6FFF	20	Reserved
400B_7000	400B_7FFF	4	OPAMP0
400B_8000	400B_BFFF	16	Reserved
400B_C000	400B_CFFF	4	PORT0
400B_D000	400B_DFFF	4	PORT1
400B_E000	400B_EFFF	4	PORT2
400B_F000	400B_FFFF	4	PORT3
400C_0000	400E_7FFF	160	Reserved
400E_8000	400E_8FFF	4	SECCON
400E_9000	400E_BFFF	12	Reserved
400E_C000	400E_CFFF	4	TRNG0
400E_D000	400E_DFFF	4	Reserved
400E_E000	400E_EFFF	4	RTC0
400E_F000	400F_2FFF	16	Reserved
400F_3000	400F_3FFF	4	GPIO0
400F_4000	400F_4FFF	4	GPIO1
400F_5000	400F_5FFF	4	GPIO2
400F_6000	400F_6FFF	4	GPIO3
400F_7000	400F_FFFF	36	Reserved
Fast peripherals			
4010_0000	4010_0FFF	4	Reserved

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table 21. Peripheral memory map (continued)

Start address (hex)	End address (hex)	Size (KB)	Peripheral instance
4010_1000	4010_1FFF	4	DBGMB0
4010_2000	4010_8FFF	28	Reserved

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Chapter 4
Bus and Memory Architecture

4.1 Bus and memory architecture

This chapter provides an overview of system bus and memory architecture of this chip. The memory architecture supports a wide range of target applications and maximizes performance while maintaining low power consumption. The memory system of the chip includes SRAM, internal flash, and external memory.

The following figure shows the system bus diagram of this chip.

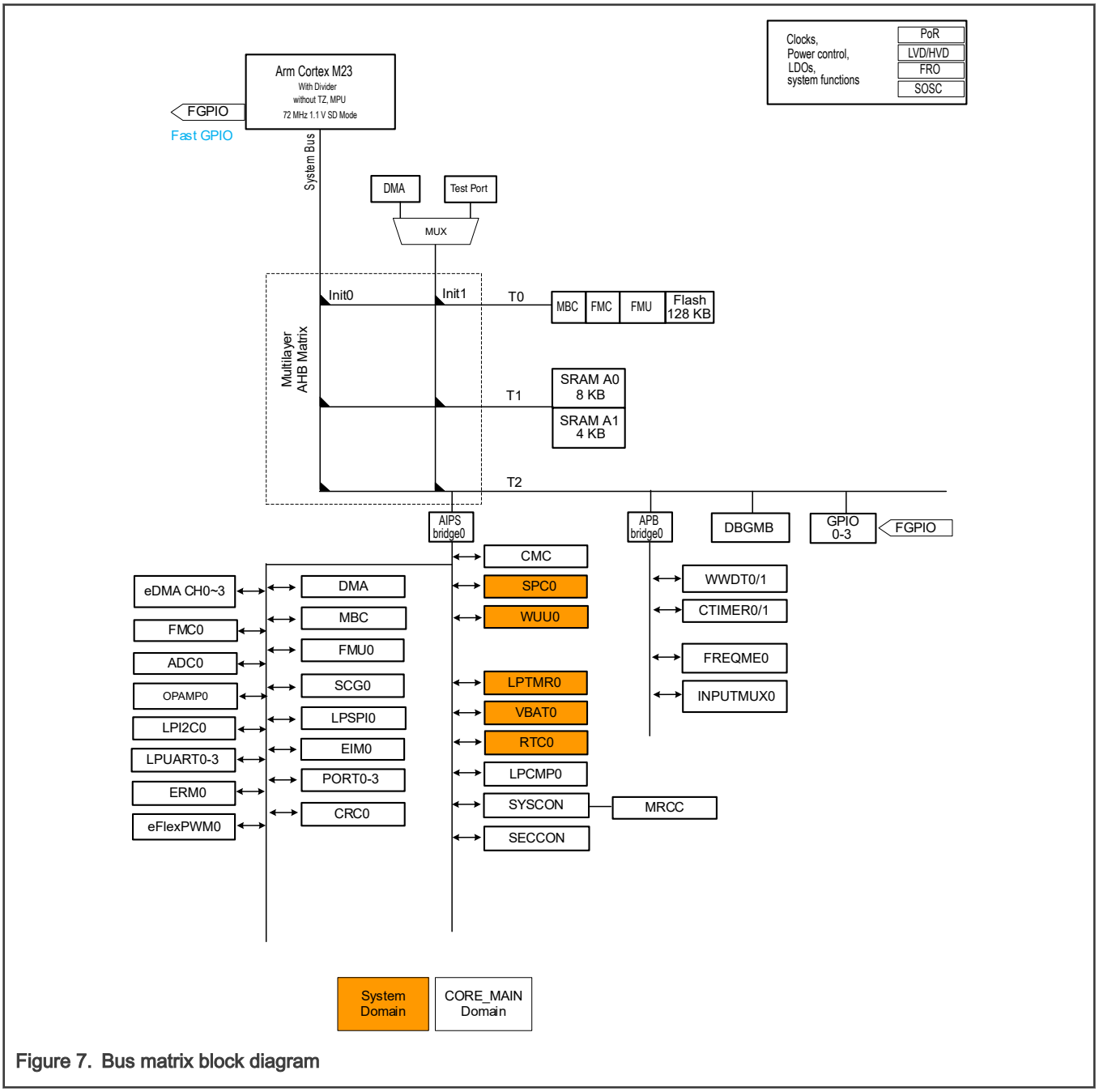


Figure 7. Bus matrix block diagram

All information provided in this document is subject to legal disclaimers.
© 2026 NXP B.V. All rights reserved.

4.2 System bus priority and arbitration

The multilayer AHB bus matrix allows for concurrent accesses when two initiators are attempting to access to different target ports. If two initiators attempt an access to the same target port at the same time then arbitration is required.

The SYSCON_AHBMATPRIO register is where the programmable priorities for each of the initiator ports can be configured. Initiators are assigned a priority value between 0 and 3 with 3 being the highest priority. If two ports have the same priority, then the lower port number is given priority.

There are some initiator ports that are shared between two initiators. Where the port is shared, only one of the initiators can have an active access at a time. The priority between two initiators sharing a port uses a PARK_ON_LAST arbitration scheme. This means that the last of the two initiators to use the port is given priority. Before either of the initiators has used the port, there is a default priority scheme that determines which initiator is prioritized for the first access. The table below lists the initiator or initiators for each of the ports.

Table 22. AHB matrix priority control register

Bit	Initiator name	Initiator AHB ID	Reset
1:0	Reserved	0x0	0x0
3:2	CM23 system bus (INIT0)	0x1	0x0
5:4	Reserved	0x2	0x0
7:6	Reserved	0x3	0x0
9:8	eDMA controller bus (INIT1)	0x5	0x0
11:10	Reserved	0x4	0x0
13:12	Reserved	0xC	0x0
15:14	Reserved	0x7	0x0
17:16	Reserved	0x8	0x0
19:18	Reserved	0x9	0x0
21:20	Reserved	0xA	0x0
23:22	Reserved	0xB	0x0
25:24	Reserved	0x6	0x0
27:26	Reserved	0xD	0x0
29:28	Reserved	0xE	0x0
31:30	Reserved	Reserved	0x0

AHB to APB Bridge

The bridge connects legacy LPC APB/VPB interface peripherals.

AIPS bridge

The bridge connects IPS/APB interface peripherals.

4.3 SRAM configuration

The chip provides up to 12 KB SRAM.

- SRAM A0 is up to 8 KB (and supports ECC)
- SRAM A1 is up to 4 KB

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

SRAM permission control

This chip provides a logic between RAM controller and AHB target port to control SRAM write permission by RAM XEN Control (RAM_XEN):

- RAMA0_XEN controls SRAM A0 8 KB.
- RAMA1_XEN controls SRAM A1 4 KB.
- A lock bit is also provided in RAM_XEN. Once locked, the register can't be programmed unless there is a system reset.

4.3.1 SRAM ECC

RAM A0 supports single-bit correction, and two bits detection ECC on every 32-bit aligned field. ECC supports byte and half-word operation. ECC can be disabled via RAM Control (RAM_CTRL).

4.3.2 RAM ECC error

The Error Reporting Module (ERM) provides information and optional interrupt notification on SRAM error events associated with ECC. The syndrome and error address information is captured along with error event in ERM registers.

The Error Injection Module (EIM) can be used to induce artificial errors in the RAM ECC. EIM can inject single-bit and multi-bit inversions on data. Injecting faults on memory accesses can be used to exercise the SEC-DED ECC function of the related system.

4.3.3 SRAM size

Table 23. SRAM size

MCX151		MCX161 and MCX162		System TCM (STCM)
Start address	End address	Start address	End address	
0x2000_0000	0x2000_17FF	0x2000_0000	0x2000_1FFF	RAM A0
—	—	0x2000_2000	0x2000_2FFF	RAM A1

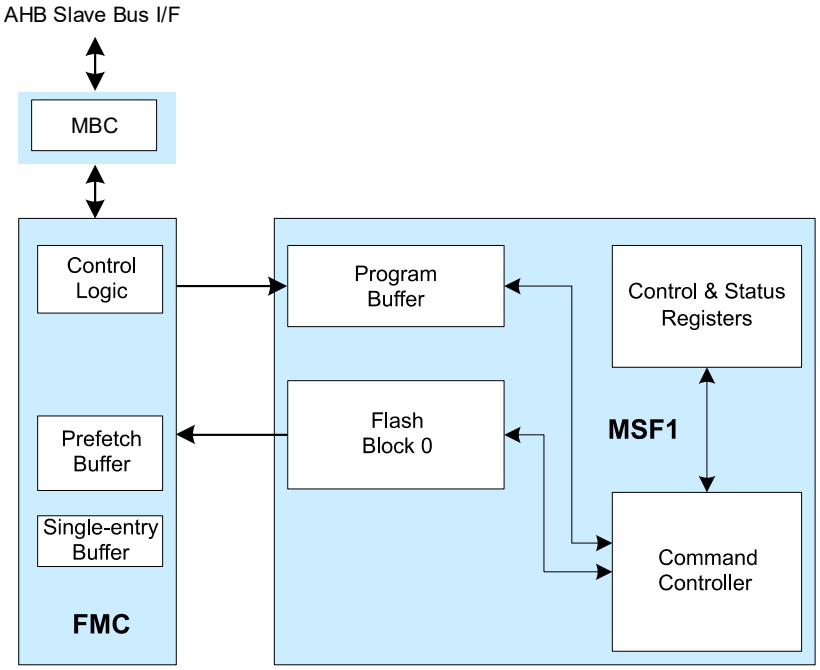
4.4 Flash subsystem

The Flash subsystem includes MBC, FMC and FMU (MSF1). FMC implements a 128-bit entry buffer and a 128-bit prefetch buffer. NVM Control (NVM_CTRL) controls the prefetch buffer.

The following figure shows the high-level block diagram of the on-chip flash.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.



Flash array configuration

This chip provides single flash array. The flash array includes 128 KB main flash, 32 KB IFR0 and 8 KB IFR1. Up to 64 KB flash starting from address 0h of main flash and 16 KB at highest address of main flash are open to customer. Part of IFR0 and IFR1 access permission is option to customer. See Table 20 and IFR configuration for details.

Flash ECC

Flash array supports ECC on every 128 bits. This chip provides NVM Control (NVM_CTRL) to disable Flash ECC error to generate bus fault, and use interrupt to handle ECC error.

Flash performance

- 72 MHz SD (Standard Drive) mode, 1 wait state

4.4.1 Flash memory controller

The Flash Memory Controller (FMC) manages accesses performed by the bus initiators of the system to the flash memory. The FMC accelerates flash memory transfers to allow program code execution at higher clock frequency than flash memory.

The FMC provides two separate mechanisms for accelerating read operations to the flash memory:

- A single-entry 128-bit buffer, which can store previously accessed flash memory
- A 128-bit prefetch buffer, which can prefetch the next 128-bit flash memory location

NOTE

The FMC has no visibility into flash memory erase and program cycles because the Flash Memory module manages them directly.

4.4.1.1 Single entry buffer

The width of the bus between the FMC and the flash blocks is 128-bit. A single read of the flash will return an entire bus width worth of data. Because the flash bus is wider than the initiator's bus, the Single-Entry Buffer is used to store data from the last flash read for quick access times on subsequent reads if the buffer is hit.

The FMC provides invalidation control for the Single-Entry Buffer.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

4.4.1.2 Prefetch buffer

When speculative reads are enabled, the FMC immediately requests the next sequential address after a read completes. The next 128-bit memory location is read and placed in the Prefetch Buffer. The speculative prefetch mechanism improves performance by reducing or even eliminating wait states when accessing sequential code and/or data.

Speculative prefetching is programmable for each bank of memory. The FMC provides invalidation control for the Prefetch Buffer.

4.5 IFR configuration

See CMPA configuration options for IFR0 CMPA configurations.

The following table provides the IFR1 configurations.

Table 24. IFR1 configurations

IFR1 address	Field name	Value	Field description
0x0110_0800 – 0x0110_080F	UUID[127:0]	Chip specific	Universal Unique Identifier (UUID) programmed per part as per RFC4122 version 5 specification.
0x0110_0810 – 0x0110_081F	ECID[127:0]	Chip specific	NXP's Electronic Chip Identifier(ECID) used to identifyi manufacturing batch the part belongs to.
0x0110_0820	TRNG_HE_CFG_MCTL	0x00010001	TRNG High Entropy Use Case Trim Value. Single oscillator mode.
0x0110_0824	TRNG_HE_CFG_SCMISC	0x20020	
0x0110_0828	TRNG_HE_CFG_SDCTL	0xC80400	
0x0110_082c	TRNG_HE_CFG_FRQMIN	0x7D0	
0x0110_0830	TRNG_HE_CFG_FRQMAX	0x61A8	
0x0110_0834	TRNG_HE_CFG_SCML	0xA90254	
0x0110_0838	TRNG_HE_CFG_SCR1L	0x7000BB	
0x0110_083C	TRNG_HE_CFG_SCR2L	0x4D0069	
0x0110_0840	TRNG_HE_CFG_SCR3L	0x400061	
0x0110_0844	TRNG_HE_CFG_SEC_CFG	0x0	
0x0110_0848	TRNG_HE_CFG_INT_CTRL	0x0	
0x0110_084C	TRNG_HE_CFG_INT_MASK	0x0	
0x0110_0850	TRNG_HE_CFG_INT_OSC2_CTL	0x00001000	

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Chapter 5
Flash Memory Module (FMU)

5.1 Chip-specific FMU information

Table 25. Reference links to related information

Topic	Related module	Reference
Full description	MSF1	Flash Memory Module
System memory map		Memory map
Clocking		Clock distribution
Power Management		Power Management
Signal multiplexing	Port control	Signal multiplexing

5.1.1 Module instances

This chip contains one instance of the FMU module, FMU0.

5.1.2 FCTRL[RWSC] Configuration recommendation

The recommended configuration of Read Wait-State Control (RWSC) field of Flash Control Register (FCTRL) is shown in below table.

Table 26. FCTRL[RWSC] configuration in SD mode and speed

AHB clock frequency (normal/middle drive)	RWSC
Frequency≤36 MHz	0000b
36 MHz < Frequency≤72 MHz	0001b
Reserved	others

5.1.3 Clocking

The following clocks are used for FMU:

Table 27. FMU clocks

Type of clock	Description
SYSTEM_CLK	Access flash array (read flash), the same as CPU clock (CPU_CLK)
SLOW_CLK	Access FMU registers
fro_12m	Program flash array

5.2 Overview

The flash module includes the following accessible memory regions:

- Program flash memory for vector space and code store
- Separate flash memory for parameter store

All information provided in this document is subject to legal disclaimers.
© 2026 NXP B.V. All rights reserved.

Flash memory is ideal for single-supply applications, permitting in-the-field erase and reprogramming operations without the need for any external high voltage power sources.

The flash module includes a command controller that executes commands to modify flash memory contents. An erased bit reads '1' and a programmed bit reads '0'. The programming operation is unidirectional; it can only move bits from the '1' state (erased) to the '0' state (programmed). Only the erase operation restores bits from '0' to '1'; bits cannot be programmed from a '0' to a '1'.

CAUTION

A flash memory location must be in the erased state before being programmed. Cumulative programming of bits (back-to-back program operations without an intervening erase) within a flash phrase or page is not allowed. Re-programming of existing 0s to 0 is not allowed as this overstresses the device.

The standard shipping condition for flash memory is erased. Data loss over time may occur due to degradation of the erased ('1') states and/or programmed ('0') states. Therefore, it is recommended that flash memory be re-erased immediately prior to factory programming to ensure that the full data retention capability is achieved.

5.2.1 Block diagram

The block diagram of the flash module is shown in the following figure.

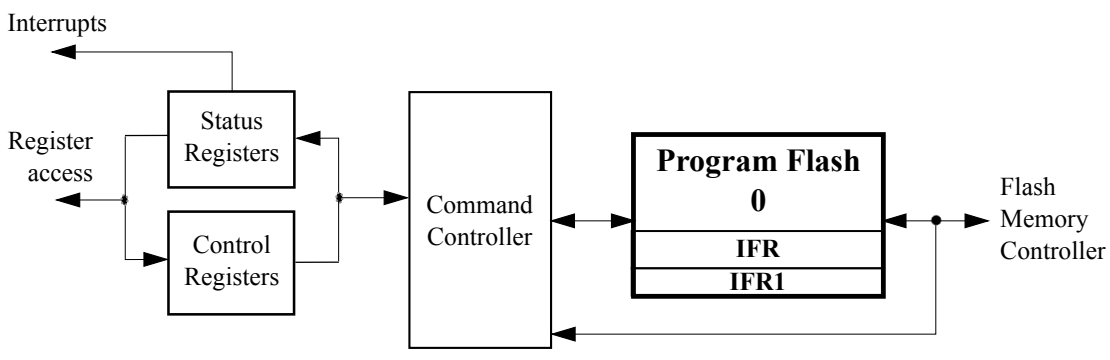


Figure 8. Flash block diagram

5.2.2 Features

The flash memory module provides the following features:

- Sector size of 8 Kbytes
- Single-bit error correction and double-bit error detection for each flash or IFR phrase
- Signature generation for flash contents
- Command-based flash program, erase, and verify operations
- Internal high-voltage supply generator for flash program and erase operations
- Optional interrupt generation upon flash command completion
- Optional interrupt generation upon double-bit error detection during flash reads from the FMC

NOTE

See the chip configuration details for the exact amount of flash memory available.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

5.3 Functional description

The following sections describe functional details of the flash module.

5.3.1 Operations

Operations are typically used to modify flash memory contents. The next sections describe:

- Command write sequence used to set flash command parameters and launch execution
- Available flash commands
- Commands impacted by Block Checker and FMC access protection

5.3.1.1 Command write sequence

Flash commands are accomplished using a command write sequence illustrated in Figure 9. The command controller monitors the transaction protection level on writes during the command write sequence, performs various checks on the command (FCCOB) content, and continues with command execution if all requirements are fulfilled.

Before launching a command, the ACCERR, PVIOL, and CMDABT flags in the FSTAT register must be zero and the CCIF flag must be one to verify that any previous command has completed. The FAIL flag in the FSTAT register is cleared by the command controller when a command is launched unless the FAIL flag was set during initialization due to FMU parameters not being loaded from IFR1. If CCIF is clear, the previous command execution is still active, a new command write sequence cannot be started, and all writes to the FCCOB registers are ignored.

5.3.1.1.1 Load the FCCOB Registers

The user must load the FCCOB registers with all parameters required by the specific flash command. The contents of any FCCOB register not used by the specific flash command are ignored except for FCCOB1 (command options) where undefined bits should be cleared. The individual registers that make up the FCCOB data set can be written in any order. All writes to the FCCOB registers in a command write sequence must occur at the same transaction protection level. Any write to the FCCOB registers that violates this rule will be ignored.

Addresses loaded into FCCOB registers assume the flash and IFR spaces start at 0000_0000h.

5.3.1.1.2 Launch the Command by Clearing CCIF

Once all relevant command parameters have been loaded, the user launches the command by clearing the FSTAT[CCIF] bit by writing a '1' to it. The write to clear CCIF must occur at the same transaction protection level as the writes used to load the FCCOB registers. Any write to clear CCIF that violates this rule will be ignored. The CCIF flag remains clear until the flash command completes.

The FSTAT register contains a blocking mechanism which prevents a new flash command from launching (unable to clear CCIF) if the previous command resulted in the setting of an access error (ACCERR), protection violation (PVIOL), or active command abort flag (CMDABT). In error scenarios, two writes to FSTAT are required to initiate the next command: the first write to clear these flags, the second write to clear CCIF. The write to clear these flags can occur at any transaction protection level.

5.3.1.1.3 Command Execution and Error Reporting

Flash command processing contains the following steps:

1. The command controller performs a series of parameter checks, if applicable, which are unique to each command. If the parameter check fails, the FSTAT[ACCERR] flag is set. ACCERR reports invalid instruction codes or options, out of bounds or misaligned addresses, or missing FMU parameters. In the case FMU parameters were not loaded from IFR1 during initialization, the ACCERR flag sets and the FSTAT[FAIL] flag remains set. Command processing never proceeds to execution when the parameter check fails. Instead, command processing is terminated after setting the FSTAT[CCIF] flag. Note that for program and erase operations, PEWEN will not set if FMU parameters were not loaded and PERDY will not set if program or erase related writes to flash or IFR space violate command requirements.
2. If necessary, the command controller performs a protection check to see if the command is allowed to execute on the requested memory space. If the protection check fails, the FSTAT[PVIOL] flag is set. Command processing never

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

proceeds to execution when the protection check fails. Instead, command processing is terminated after setting the FSTAT[CCIF] flag.

3. If the parameter and protection checks pass, the command proceeds to execution. Run-time errors, such as failure to verify, may occur during the execution phase and are reported in the FAIL flag. A flash command may have access errors, protection violations, and run-time errors, but the run-time errors are not seen until all access errors and protection violations have been corrected.
4. If FCTRL[ABTREQ] sets during command execution, the operation will abort with the FSTAT[CMDABT] flag set.
5. Command execution results, if applicable, are reported back to the user via the FCCOB registers.
6. The command controller sets FSTAT[CCIF] signifying that the command has completed.

CAUTION

If the FAIL flag is set along with ACCERR, FMU parameters were not loaded from IFR1 during initialization. In this case, it is recommended that the flash module be reinitialized. If the issue persists, flash commands will continue to set the ACCERR and FAIL flags.

The flow for a generic command write sequence is illustrated in the following figure.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

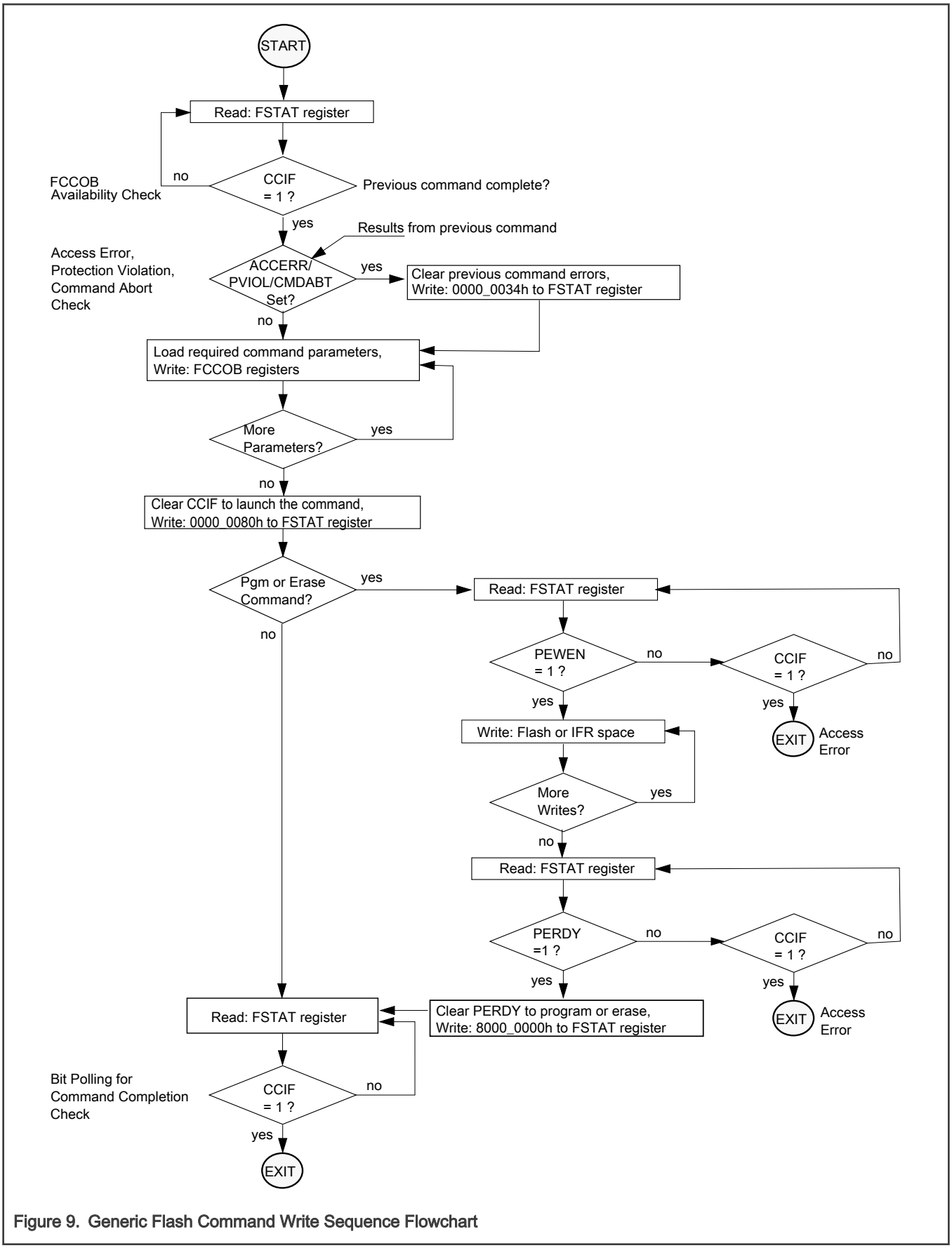


Figure 9. Generic Flash Command Write Sequence Flowchart

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

5.3.1.1.4 Aborting a Command Write Sequence or Command Operation

FSTAT[CMDP] sets at the start of a command write sequence, typically by writing to one of the FCCOB registers. With CMDP set, setting the FCTRL[ABTREQ] bit allows a user to abort a command write sequence and take control over a new command write sequence. The ABTREQ bit can only be set by a user with the same domain ID as indicated by FSTAT[CMDDID] and at the same or higher security/privilege protection level when compared to FSTAT[CMDPRT], the current owner of the command write sequence. While ABTREQ is set, CCIF cannot be cleared to launch a command.

If FSTAT[CCIF] is high when ABTREQ is set, the command write sequence will abort and ownership transferred to the user setting the ABTREQ bit. The FSTAT[CWSABT] flag will set and the CMDPRT field will be updated based on the write transaction that set the ABTREQ bit. The ABTREQ bit will then be cleared automatically.

If CCIF is low when ABTREQ is set, the command controller will abort execution of the current command operation, set FSTAT[CMDABT], and set CCIF. The CMDPRT field will be updated based on the write transaction that set the ABTREQ bit. The ABTREQ bit will then be cleared automatically. While CMDABT is set, CCIF cannot be cleared to launch a command. Aborting a program or erase operation may leave the flash or IFR space being modified in an indeterminate state.

NOTE

Aborting certain flash commands may take longer to abort than to execute. See the device data sheet for flash command timing specifications.

5.3.1.2 Flash commands

The following table summarizes the function of all flash commands. If any column is marked with an 'X', the flash command is relevant to that particular memory resource.

FCMD	Command	Program flash 0	IFR	Function
00h	Read 1s All (RD1ALL)	x	x	Verify that all flash and IFR space is erased
01h	Read 1s Block (RD1BLK)	x		Verify that a flash block is erased
02h	Read 1s Sector (RD1SCR)	x		Verify that a flash sector is erased
03h	Read 1s Page (RD1PG)	x		Verify that a flash page is erased
04h	Read 1s Phrase (RD1PHR)	x		Verify that a flash phrase is erased
05h	Read into MISR (RDMISR)	x		Generate MISR signature for range of flash pages
12h	Read 1s IFR Sector (RD1ISCR)		x	Verify that an IFR sector is erased
13h	Read 1s IFR Page (RD1IPG)		x	Verify that an IFR page is erased
14h	Read 1s IFR Phrase (RD1IPHR)		x	Verify that an IFR phrase is erased
15h	Read IFR into MISR (RDIMISR)		x	Generate MISR signature for range of IFR pages

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

FCMD	Command	Program flash 0	IFR	Function
23h	Program Page (PGMPG)	x	x	Program data to a flash or IFR page
24h	Program Phrase (PGMPHR)	x	x	Program data to a flash or IFR phrase
42h	Erase Sector (ERSSCR)	x	x	Erase a flash sector

5.3.1.3 Flash commands impacted by Block Checker and FMC access protection

The following flash commands are impacted by Block Checker and FMC access protection:

- Program Phrase (PGMPHR)
- Program Page (PGMPG)
- Erase Sector (ERSSCR)

During execution of these flash commands, a sequence of AHB writes to flash or IFR space are required with validity checked by the Block Checker or FMC. When writes are enabled by the FMU, violations detected by the FMC are reported as an access error in the FMU. The other violations are handled by the Block Checker or FMC.

Table 28. Flash Write Access Checks

Flash Write Violation Condition	Block Checker	FMC	FMU
Flash or IFR address out-of-range	X		
Flash or IFR write permission check fails	X		
Write to IFR1 space		X	
Write width not 32-bit		X	
Writes not enabled by FSTAT[PEWEN]		X	
First write in sequence not properly aligned per FSTAT[PEWEN]			X
Non-sequential write after first aligned write in sequence			X
Bus permissions for the AHB write do not match the bus permissions of the APB write that launched the flash command			X

5.3.2 Flash command descriptions

This section describes all flash commands that can be launched by a valid command write sequence. The flash module sets the FSTAT[ACCERR] bit and aborts the command execution if any of the following illegal conditions occur:

- There is an unrecognized command code in the FCCOB FCMD field.
- There is an error in an FCCOB field for the specific command. Refer to the error handling table provided for each command.

The ACCERR, PVIOL, and CMDABT flags in the FSTAT register should be cleared prior to starting a command write sequence. As described in Launch the Command by Clearing CCIF, a new command cannot be launched while these flags are set.

Do not attempt to read a flash block while a flash command is running (CCIF = 0) on that same block. The flash module may return a transfer error to the FMC resulting in a bus fault. The exception to this rule is during program operations where the flash block is still readable after the command is launched up until the time the FSTAT[PERDY] flag is cleared.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

CAUTION

Flash data must be in the erased state before being programmed. Cumulative programming of bits (adding more zeros) is not allowed.

5.3.2.1 Read 1s All Command

The Read 1s All command checks if all flash and IFR space are in the erased state.

Table 29. Read 1s All Command FCCOB Requirements

FCCOB Number	FCCOB Contents
0	00h (RD1ALL)

Upon clearing CCIF to launch the Read 1s All command, the command controller:

- verifies that all flash and IFR space are in the erased state

If all flash and IFR space are not in the erased state, the FSTAT[FAIL] flag is set. The CCIF flag sets after the Read 1s All operation completes.

Table 30. Read 1s All Command Error Handling

Error Condition	Error Bit
FMU parameters not loaded from IFR1 during initialization	FSTAT[ACCERR]
Operation is aborted	FSTAT[CMDABT]
FMU parameters not loaded from IFR1 during initialization	FSTAT[FAIL]
Verify fails	FSTAT[FAIL]

5.3.2.2 Read 1s Block Command

The Read 1s Block command checks if a flash block is in the erased state.

Table 31. Read 1s Block Command FCCOB Requirements

FCCOB Number	FCCOB Contents
0	01h (RD1BLK)
2	Block address

Upon clearing CCIF to launch the Read 1s Block command, the command controller:

- verifies that the selected flash block is in the erased state

If the selected flash block is not in the erased state, the FSTAT[FAIL] flag is set. The CCIF flag sets after the Read 1s Block operation completes.

Table 32. Read 1s Block Command Error Handling

Error Condition	Error Bit
FMU parameters not loaded from IFR1 during initialization	FSTAT[ACCERR]
Address is not valid	FSTAT[ACCERR]
Address is not block aligned	FSTAT[ACCERR]
Operation is aborted	FSTAT[CMDABT]

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table 32. Read 1s Block Command Error Handling (continued)

Error Condition	Error Bit
FMU parameters not loaded from IFR1 during initialization	FSTAT[FAIL]
Verify fails	FSTAT[FAIL]

5.3.2.3 Read 1s Sector Command

The Read 1s Sector command checks if a flash sector is in the erased state.

Table 33. Read 1s Sector Command FCCOB Requirements

FCCOB Number	FCCOB Contents
0	02h (RD1SCR)
2	Sector address

Upon clearing CCIF to launch the Read 1s Sector command, the command controller:

- verifies that the selected flash sector is in the erased state

If the selected flash sector is not in the erased state, the FSTAT[FAIL] flag is set. The CCIF flag sets after the Read 1s Sector operation completes.

Table 34. Read 1s Sector Command Error Handling

Error Condition	Error Bit
FMU parameters not loaded from IFR1 during initialization	FSTAT[ACCERR]
Address is not valid	FSTAT[ACCERR]
Address is not sector aligned	FSTAT[ACCERR]
Operation is aborted	FSTAT[CMDABT]
FMU parameters not loaded from IFR1 during initialization	FSTAT[FAIL]
Verify fails	FSTAT[FAIL]

5.3.2.4 Read 1s Page Command

The Read 1s Page command checks if a flash page is in the erased state.

Table 35. Read 1s Page Command FCCOB Requirements

FCCOB Number	FCCOB Contents
0	03h (RD1PG)
2	Page address

Upon clearing CCIF to launch the Read 1s Page command, the command controller:

- verifies that the selected flash page is in the erased state

If the selected flash page is not in the erased state, the FSTAT[FAIL] flag is set. The CCIF flag sets after the Read 1s Page operation completes.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table 36. Read 1s Page Command Error Handling

Error Condition	Error Bit
FMU parameters not loaded from IFR1 during initialization	FSTAT[ACCERR]
Address is not valid	FSTAT[ACCERR]
Address is not page aligned	FSTAT[ACCERR]
Operation is aborted	FSTAT[CMDABT]
FMU parameters not loaded from IFR1 during initialization	FSTAT[FAIL]
Verify fails	FSTAT[FAIL]

5.3.2.5 Read 1s Phrase Command

The Read 1s Phrase command checks if a flash phrase is in the erased state.

Table 37. Read 1s Phrase Command FCCOB Requirements

FCCOB Number	FCCOB Contents
0	04h (RD1PHR)
2	Phrase address

Upon clearing CCIF to launch the Read 1s Phrase command, the command controller:

- verifies that the selected flash phrase is in the erased state

If the selected flash phrase is not in the erased state, the FSTAT[FAIL] flag is set. The CCIF flag sets after the Read 1s Phrase operation completes.

Table 38. Read 1s Phrase Command Error Handling

Error Condition	Error Bit
FMU parameters not loaded from IFR1 during initialization	FSTAT[ACCERR]
Address is not valid	FSTAT[ACCERR]
Address is not phrase aligned	FSTAT[ACCERR]
Operation is aborted	FSTAT[CMDABT]
FMU parameters not loaded from IFR1 during initialization	FSTAT[FAIL]
Verify fails	FSTAT[FAIL]

5.3.2.6 Read into MISR command

The Read into MISR operation generates a signature based on the contents of the selected flash memory using an embedded MISR.

Table 39. Read into MISR Command FCCOB Requirements

FCCOB Number	FCCOB Contents
0	05h (RDMISR) [7:0]
2	Starting flash page address

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table 39. Read into MISR Command FCCOB Requirements (continued)

FCCOB Number	FCCOB Contents
3	Ending flash phrase address
4	Word 0 of MISR seed
5	Word 1 of MISR seed
6	Word 2 of MISR seed
7	Word 3 of MISR seed
Returned values	
4	Word 0 of MISR signature
5	Word 1 of MISR signature
6	Word 2 of MISR signature
7	Word 3 of MISR signature

Upon clearing CCIF to launch the Read into MISR command, the command controller:

1. initializes the MISR with the seed provided
2. processes flash data starting with the starting flash page address provided
3. continues until the ending flash phrase address has been processed (must be the last phrase in a page)
4. returns the resulting signature into the FCCOB register bank unless an uncorrectable ECC fault is detected

The CCIF flag sets after the Read into MISR operation completes. If the operation is aborted by setting FCTRL[ABTREQ], a signature will not be returned if the abort occurs prior to the operation writing the signature.

MISR polynomial is $X^{128} + X^{126} + X^{101} + X^{99} + 1$.

Table 40. Read into MISR Command Error Handling

Error Condition	Error Bit
FMU parameters not loaded from IFR1 during initialization	FSTAT[ACCERR]
Address is not valid	FSTAT[ACCERR]
Starting address is not flash page aligned	FSTAT[ACCERR]
Ending address is not flash phrase aligned	FSTAT[ACCERR]
Ending address is not the last phrase in a page	FSTAT[ACCERR]
Starting address is larger than the ending address	FSTAT[ACCERR]
Requested range crosses a flash block boundary	FSTAT[ACCERR]
Operation is aborted	FSTAT[CMDABT]
FMU parameters not loaded from IFR1 during initialization	FSTAT[FAIL]
Uncorrectable ECC error is detected	FSTAT[FAIL]

5.3.2.7 Read 1s IFR Sector Command

The Read 1s IFR Sector command checks if an IFR sector is in the erased state.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table 41. Read 1s IFR Sector Command FCCOB Requirements

FCCOB Number	FCCOB Contents
0	12h (RD1ISCR)
2	IFR sector address

Upon clearing CCIF to launch the Read 1s IFR Sector command, the command controller:

- verifies that the selected IFR sector is in the erased state

If the selected IFR sector is not in the erased state, the FSTAT[FAIL] flag is set. The CCIF flag sets after the Read 1s IFR Sector operation completes.

Table 42. Read 1s IFR Sector Command Error Handling

Error Condition	Error Bit
FMU parameters not loaded from IFR1 during initialization	FSTAT[ACCERR]
Address is not valid	FSTAT[ACCERR]
Address is not IFR sector aligned	FSTAT[ACCERR]
Operation is aborted	FSTAT[CMDABT]
FMU parameters not loaded from IFR1 during initialization	FSTAT[FAIL]
Verify fails	FSTAT[FAIL]

5.3.2.8 Read 1s IFR Page Command

The Read 1s IFR Page command checks if an IFR page is in the erased state.

Table 43. Read 1s IFR Page Command FCCOB Requirements

FCCOB Number	FCCOB Contents
0	13h (RD1IPG)
2	IFR page address

Upon clearing CCIF to launch the Read 1s IFR Page command, the command controller:

- verifies that the selected IFR page is in the erased state

If the selected IFR page is not in the erased state, the FSTAT[FAIL] flag is set. The CCIF flag sets after the Read 1s IFR Page operation completes.

Table 44. Read 1s IFR Page Command Error Handling

Error Condition	Error Bit
FMU parameters not loaded from IFR1 during initialization	FSTAT[ACCERR]
Address is not valid	FSTAT[ACCERR]
Address is not IFR page aligned	FSTAT[ACCERR]
Operation is aborted	FSTAT[CMDABT]
FMU parameters not loaded from IFR1 during initialization	FSTAT[FAIL]
Verify fails	FSTAT[FAIL]

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

5.3.2.9 Read 1s IFR Phrase Command

The Read 1s IFR Phrase command checks if an IFR phrase is in the erased state.

Table 45. Read 1s IFR Phrase Command FCCOB Requirements

FCCOB Number	FCCOB Contents
0	14h (RD1IPHR)
2	IFR phrase address

Upon clearing CCIF to launch the Read 1s IFR Phrase command, the command controller:

- verifies that the selected IFR phrase is in the erased state

If the selected IFR phrase is not in the erased state, the FSTAT[FAIL] flag is set. The CCIF flag sets after the Read 1s IFR Phrase operation completes.

Table 46. Read 1s IFR Phrase Command Error Handling

Error Condition	Error Bit
FMU parameters not loaded from IFR1 during initialization	FSTAT[ACCERR]
Address is not valid	FSTAT[ACCERR]
Address is not IFR phrase aligned	FSTAT[ACCERR]
Operation is aborted	FSTAT[CMDABT]
FMU parameters not loaded from IFR1 during initialization	FSTAT[FAIL]
Verify fails	FSTAT[FAIL]

5.3.2.10 Read IFR into MISR command

The Read IFR into MISR operation generates a signature based on the contents of the selected IFR space using an embedded MISR.

Table 47. Read IFR into MISR Command FCCOB Requirements

FCCOB Number	FCCOB Contents
0	15h (RDIMISR) [7:0]
2	Starting IFR page address
3	Ending IFR phrase address
4	Word 0 of MISR seed
5	Word 1 of MISR seed
6	Word 2 of MISR seed
7	Word 3 of MISR seed
Returned values	
4	Word 0 of MISR signature
5	Word 1 of MISR signature
6	Word 2 of MISR signature
7	Word 3 of MISR signature

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Upon clearing CCIF to launch the Read IFR into MISR command, the command controller:

1. initializes the MISR with the seed provided
2. processes IFR data starting with the starting IFR page address provided
3. continues until the ending IFR phrase has been processed (must be the last phrase in a page)
4. returns the resulting signature into the FCCOB register bank unless an uncorrectable ECC fault is detected

The CCIF flag sets after the Read IFR into MISR operation completes. If the operation is aborted by setting FCTRL[ABTREQ], a signature will not be returned if the abort occurs prior to the operation writing the signature.

MISR polynomial is $X^{128} + X^{126} + X^{101} + X^{99} + 1$.

Table 48. Read IFR into MISR Command Error Handling

Error Condition	Error Bit
FMU parameters not loaded from IFR1 during initialization	FSTAT[ACCERR]
Address is not valid	FSTAT[ACCERR]
Starting address is not IFR page aligned	FSTAT[ACCERR]
Ending address is not IFR phrase aligned	FSTAT[ACCERR]
Ending address is not the last phrase in a page	FSTAT[ACCERR]
Starting address is larger than the ending address	FSTAT[ACCERR]
Operation is aborted	FSTAT[CMDABT]
FMU parameters not loaded from IFR1 during initialization	FSTAT[FAIL]
Uncorrectable ECC error is detected	FSTAT[FAIL]

5.3.2.11 Program Page command

The Program Page operation programs 32 words of data to a previously erased flash or IFR page.

CAUTION

The page must be in the erased state before being programmed. Cumulative programming of bits (back-to-back program operations without an intervening erase) within a page is not allowed.

Table 49. Program Page Command FCCOB Requirements

FCCOB Number	FCCOB Contents
0	23h (PGMPG) [7:0]

Upon clearing CCIF to launch the Program Page command, the command controller:

1. sets FSTAT[PEWEN] to enable writes to the flash and IFR space in the system memory map
2. waits for the user to write 32 consecutive words to the flash or IFR space with the first write being page aligned
3. clears FSTAT[PEWEN] field
4. sets FSTAT[PERDY]
5. waits for the user to clear FSTAT[PERDY]
6. programs the data into the targeted page written to in step 2
7. verifies bits to be programmed are programmed to the program verify level

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

The CCIF flag sets after the Program Page operation completes. The Program Page operation can be aborted by setting FCTRL[ABTREQ]. If the operation is aborted, the operation terminates with the FSTAT[CMDABT] flag set. Once CCIF is set, the Program Page command may be repeated (same addresses, same data) to complete the program operation without erasing the sector containing the targeted page. Invalid writes to flash or IFR space based on Block Checker or FMC access protection are trapped outside of the flash module.

Table 50. Program Page Command Error Handling

Error Condition	Error Bit
FMU parameters not loaded from IFR1 during initialization	FSTAT[ACCERR]
Protection level of the writes to flash or IFR space do not match the protection level of the command write sequence	FSTAT[ACCERR]
Initial address is not page aligned and subsequent addresses phrase aligned	FSTAT[ACCERR]
Number of writes to flash or IFR space is more than expected	FSTAT[ACCERR]
Operation is aborted	FSTAT[CMDABT]
FMU parameters not loaded from IFR1 during initialization	FSTAT[FAIL]
Verify fails	FSTAT[FAIL]

5.3.2.11.1 Page programming

The standard process of programming one or more flash or IFR pages is as follows:

1. If required, launch the Erase Sector command to erase the flash or IFR sector to be programmed (repeat for additional sectors).
2. Launch the Program Page command to enable writes to the flash space.
3. After the command controller sets FSTAT[PEWEN], write 32 consecutive words to the flash space with the first word page aligned. The data written to the flash space is not readable until the program operation has successfully completed.
4. After the flash page has been written, the command controller will stall and set FSTAT[PERDY].
5. Clear PERDY allowing the command controller to resume and program the 32 words into the flash or IFR space written to in step 3.
6. Wait for CCIF to set indicating the program operation has completed.
7. To program additional flash or IFR pages in previously erased flash memory, repeat steps 2 through 6.

5.3.2.12 Program Phrase command

The Program Phrase operation programs 4 words of data to a previously erased flash or IFR phrase.

CAUTION

The phrase must be in the erased state before being programmed. Cumulative programming of bits (back-to-back program operations without an intervening erase) within a phrase is not allowed.

Table 51. Program Phrase Command FCCOB Requirements

FCCOB Number	FCCOB Contents
0	24h (PGMPHR) [7:0]

Upon clearing CCIF to launch the Program Phrase command, the command controller:

1. sets FSTAT[PEWEN] to enable writes to the flash and IFR space in the system memory map

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

2. waits for the user to write 4 consecutive words to the flash or IFR space with the first write being phrase aligned
3. clears FSTAT[PEWEN] field
4. sets FSTAT[PERDY]
5. waits for the user to clear FSTAT[PERDY]
6. programs the data into the targeted phrase written to in step 2.
7. verifies bits to be programmed are programmed to the program verify level

The CCIF flag sets after the Program Phrase operation completes. The Program Phrase operation can be aborted by setting FCTRL[ABTREQ]. If the operation is aborted, the operation terminates with the FSTAT[CMDABT] flag set. Once CCIF is set and the CMDABT flag is cleared, the Program Phrase command may be repeated (same addresses, same data) to complete the program operation without erasing the sector containing the targeted phrase. Invalid writes to flash or IFR space based on Block Checker or FMC access protection are trapped outside of the flash module.

Table 52. Program Phrase Command Error Handling

Error Condition	Error Bit
FMU parameters not loaded from IFR1 during initialization	FSTAT[ACCERR]
Protection level of the writes to flash or IFR space do not match the protection level of the command write sequence	FSTAT[ACCERR]
Address is not phrase aligned	FSTAT[ACCERR]
Number of writes to flash or IFR space is more than expected	FSTAT[ACCERR]
Operation is aborted	FSTAT[CMDABT]
FMU parameters not loaded from IFR1 during initialization	FSTAT[FAIL]
Verify fails	FSTAT[FAIL]

5.3.2.12.1 Phrase programming

The standard process of programming one or more flash or IFR phrases is as follows:

1. If required, launch the Erase Sector command to erase the flash or IFR sector to be programmed (repeat for additional sectors).
2. Launch the Program Phrase command to enable writes to the flash and IFR space.
3. After the command controller sets FSTAT[PEWEN], write 4 consecutive words to the flash or IFR space with the first word phrase aligned. The data written to the flash or IFR space is not readable until the program operation has successfully completed.
4. After the flash or IFR phrase has been written, the command controller will stall the operation and set FSTAT[PERDY].
5. Clear PERDY allowing the command controller to resume and program the 4 words into the flash or IFR space written to in step 3.
6. Wait for CCIF to set indicating the program operation has completed.
7. To program additional flash or IFR phrases in previously erased flash memory, repeat steps 2 through 6.

5.3.2.13 Erase Sector command

The Erase Sector operation erases the IFR sector to prepare the sector for programming.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table 53. Erase Sector Command FCCOB Requirements

FCCOB Number	FCCOB Contents
0	42h (ERSSCR) [7:0]

Upon clearing CCIF to launch the Erase Sector command, the command controller:

1. sets FSTAT[PEWEN] to enable writes to the flash and IFR space in the system memory map
2. waits for the user to write 4 consecutive words to the flash or IFR space with the first write being phrase (or sector) aligned
3. clears FSTAT[PEWEN] field
4. sets FSTAT[PERDY]
5. waits for the user to clear FSTAT[PERDY]
6. erases the selected flash or IFR sector
7. verifies the selected flash or IFR sector is erased

If the selected flash or IFR sector is not in the erased state, FSTAT[FAIL] is set. The CCIF flag sets after the Erase Sector operation completes. The Erase Sector operation can be aborted by setting FCTRL[ABTREQ]. If the operation is aborted, the operation terminates with the FSTAT[CMDABT] flag set. Once CCIF is set, the Erase Sector command may be repeated to complete the erase operation. Invalid writes to flash or IFR space based on Block Checker or FMC access protection are trapped outside of the flash module.

Table 54. Erase Sector Command Error Handling

Error Condition	Error Bit
FMU parameters not loaded from IFR1 during initialization	FSTAT[ACCERR]
Protection level of the writes to flash or IFR space do not match the protection level of the command write sequence	FSTAT[ACCERR]
Address is not flash or IFR phrase aligned	FSTAT[ACCERR]
Operation is aborted	FSTAT[CMDABT]
IFR Sector is protected (see FCNFG register)	FSTAT[PVIOL]
FMU parameters not loaded from IFR1 during initialization	FSTAT[FAIL]
Verify fails	FSTAT[FAIL]

5.3.3 Modes

5.3.3.1 Sleep mode

If the CPU enters sleep mode while a flash command is executing, the flash module can wake the CPU via the command complete interrupt (see Interrupts).

5.3.3.2 Non-Active Power mode

If a flash command is active (CCIF = 0) when the MCU requests a non-active power mode, the command execution completes before the MCU is allowed to enter the non-active power mode.

5.3.4 Clocking

This module has no clocking considerations.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

5.3.5 Interrupts

The flash module can generate interrupt requests to the MCU upon the occurrence of various flash module events. These interrupt events and their associated status and control bits are shown in the following table.

Table 55. Flash Module Interrupt Sources

Flash Event	Readable Status Bit	Interrupt Enable Bit
Flash Command Complete	FSTAT[CCIF]	FCNFG[CCIE]
Flash ECC Fault Detected	FSTAT[DFDIF]	FCNFG[DFDIE]

NOTE

Vector addresses and their relative interrupt priority are determined at the MCU level.

5.4 External signals

The flash module contains no signals that connect off-chip.

5.5 Initialization

On each reset, the command controller initializes the following based on IFR1 content:

1. FCTRL[RWSC]
2. FMU Parameters (FMUPARM)

The FSTAT[CCIF] flag is cleared throughout the initialization flow. The flash module also holds off all access to flash registers and flash memory during initialization. Completion of the initialization flow is marked by setting FSTAT[CCIF]. If an uncorrectable ECC error is detected during initialization, the FSTAT[FAIL] flag sets. The FAIL flag also sets if FMU parameters are not loaded from IFR1 during initialization. In this case, attempts to launch a flash command terminate with the ACCERR and FAIL flags set. The recommendation is to refrain from launching flash commands if the FAIL flag sets after initialization.

If a reset occurs while any flash command is in progress, that command is immediately aborted. The state of the flash memory or IFR space being programmed or erased is not guaranteed. Command operations do not automatically resume after exiting reset.

5.6 Memory map and registers

This section describes the flash memory map for IFR/IFR1 space and registers. See the MCU system memory map for the location of all flash memory space and registers. Data read from unimplemented memory space in the flash module is undefined.

5.6.1 IFR description

Each flash block contains IFR space separate from the main flash memory. IFR space is memory mapped (see the MCU system memory map). IFR space is divided into IFR pages programmable using the Program Page command. IFR pages are further divided into IFR phrases programmable using the Program Phrase command. Once programmed, IFR phrases can be reprogrammed after a successful erase using the Erase Sector command operation unless the IFR sector is protected by the FCNFG register.

Table 56. IFR map

Offset Byte Address	IFR Field Size (Bytes)
Block 0 IFR Sector 0	

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table 56. IFR map (continued)

Offset Byte Address	IFR Field Size (Bytes)
0000h-1FFFh	8,192
Block 0 IFR Sector 1	
2000h-3FFFh	8,192
Block 0 IFR Sector 2	
4000h-5FFFh	8,192
Block 0 IFR Sector 3	
6000h-7FFFh	8,192

Each flash block also contains IFR1 space separate from the main flash memory. IFR1 space in program flash block 0 is memory mapped for read-only access (see the MCU system memory map). IFR1 space cannot be erased or programmed by the user.

Table 57. IFR1 map

Offset Byte Address	IFR1 Field Size (Bytes)
Block 0 IFR1 Sector	
0000h-15FFh	5,632
1600h-16FFh	256
1700h-1FFFh	2,304

5.6.2 Register descriptions

The flash module contains a set of memory-mapped control and status registers.

NOTE

While a command is running (FSTAT[CCIF]=0), register writes are allowed to FSTAT[PERDY,DFDIF,CWSABT], FCNFG, and FCTRL[ABTREQ,FDFD,RWSC].

5.6.2.1 Flash register descriptions

5.6.2.1.1 Flash memory map

FMU0 base address: 4009_5000h

Offset	Register	Width (In bits)	Access	Reset value
0h	Flash Status Register (FSTAT)	32	RW	See section
4h	Flash Configuration Register (FCNFG)	32	RW	See section
8h	Flash Control Register (FCTRL)	32	RW	See section
10h	Flash Common Command Object Registers (FCCOB0)	32	RW	0000_0000h
14h	Flash Common Command Object Registers (FCCOB1)	32	RW	0000_0000h

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Offset	Register	Width (In bits)	Access	Reset value
18h	Flash Common Command Object Registers (FCCOB2)	32	RW	0000_0000h
1Ch	Flash Common Command Object Registers (FCCOB3)	32	RW	0000_0000h
20h	Flash Common Command Object Registers (FCCOB4)	32	RW	0000_0000h
24h	Flash Common Command Object Registers (FCCOB5)	32	RW	0000_0000h
28h	Flash Common Command Object Registers (FCCOB6)	32	RW	0000_0000h
2Ch	Flash Common Command Object Registers (FCCOB7)	32	RW	0000_0000h

5.6.2.1.2 Flash Status Register (FSTAT)

Offset

Register	Offset
FSTAT	0h

Function

The FSTAT register reports the operational status of the flash module.

NOTE

When set, ACCERR, PVIOL, and CMDABT flags prevent the launch of any more commands until the flag is cleared (by writing a one to it).

CAUTION

When clearing the DFDIF flag, be careful not to use read-modify-write on the entire FSTAT register and inadvertently clear CCIF to launch a command.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	PERDY	0					PEWEN		0						SALV_US...	DFDIF
W	W1C															W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CMDDID				CMDP	0	CMDPRT		CCIF	CWSABT	ACCERR	PVIOL	0	CMDABT	0	FAIL
W									W1C	W1C	W1C	W1C		W1C		
Reset	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	u

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Fields

Field	Function
31 PERDY	Program-Erase Ready Control/Status Flag The PERDY flag is set by the command controller when a program or sector erase command operation has successfully completed the write phase. The command controller will stall the command operation until the user clears the PERDY flag by writing a 1 to PERDY. Failure to clear PERDY will leave the command controller stalled but the flash or IFR space targeted by the command operation will remain readable by the FMC. If FCTRL[ABTREQ] is set while the PERDY flag is set, the command controller will clear PERDY, abort the command operation, and set both CCIF and CMDABT. 0b - Program or sector erase command operation not stalled 1b - Program or sector erase command operation ready to execute
30-26 —	Reserved
25-24 PEWEN	Program-Erase Write Enable Control During program or sector erase command operations, writes are enabled to flash or IFR space in the system memory map. For phrase programming or sector erase, writes are enabled for 4 consecutive words (16 bytes) with address phrase aligned. For page programming, writes are enabled for 8 consecutive phrases (128 bytes) with address page aligned. For programming, data written to erased flash or IFR space for program commands is not available until the program operation has successfully completed. PEWEN will not assert if FMU parameters were not loaded from IFR1 during initialization. 00b - Writes are not enabled 01b - Writes are enabled for one flash or IFR phrase (phrase programming, sector erase) 10b - Writes are enabled for one flash or IFR page (page programming) 11b - Reserved
23-18 —	Reserved
17 SALV_USED	Salvage Used for Erase operation This flag indicates salvage was used during the last Erase Sector, Erase All, or Mass Erase operation and is valid at the end of the operation. Salvage involves using ECC to pass erase verify allowing single-bit faults per phrase verified. This flag remains valid until any flash command or Mass Erase operation launches at which time the flag clears. Use of salvage does not guarantee that the erase operation succeeds but only that salvage was used on at least one phrase. 0b - Salvage not used during last operation 1b - Salvage used during the last erase operation
16 DFDIF	Double Bit Fault Detect Interrupt Flag The DFDIF flag indicates an uncorrectable ECC fault was detected during a valid flash read access from the FMC. The DFDIF flag is cleared by writing a 1 to DFDIF. Writing a 0 to DFDIF has no effect. 0b - Double bit fault not detected during a valid flash read access

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	1b - Double bit fault detected (or FCTRL[FDFD] is set) during a valid flash read access
15-12 CMDDID	Command domain ID While the CMDP flag is set, the CMDDID bits reflect the domain ID of the write transaction controlling a command write sequence or aborting a command operation. Writes to load an FCCOB register, clear CCIF, or clear PERDY must have the same domain ID as the write transaction that started the command write sequence. The CMDDID bits persist after the CMDP flag clears until the CMDP flag sets again at the start of a new command write sequence. Writes to clear the CWSABT, ACCERR, PVIOL, or CMDABT flags are not influenced by the state of the CMDDID bits.
11 CMDP	Command protection status flag The CMDP flag sets when a command write sequence starts with a write to load an existing FCCOB register or clear CCIF while ACCERR, PVIOL, and CMDABT are clear. Clearing the CWSABT, ACCERR, PVIOL, or CMDABT flags as part of a command write sequence will not set the CMDP flag. While the CMDP flag is set, the protection level of any subsequent write transaction to load an FCCOB register, clear CCIF, or clear PERDY must match the CMDPRT bits and the domain ID must match the CMDDID bits. The CMDP flag remains set until cleared when CCIF is set after a command operation or Power Down recovery completes unless the CMDABT flag is set. The CMDP flag is cleared even if the operation ends with ACCERR or PVIOL set. When the CMDP flag is cleared, the CMDPRT and CMDDID bits remain unchanged. The CMDP flag is also cleared by a mass erase request with CCIF set. 0b - Command protection level and domain ID are stale 1b - Command protection level (CMDPRT) and domain ID (CMDDID) are set
10 —	Reserved
9-8 CMDPRT	Command protection level While the CMDP flag is set, the CMDPRT bits reflect the secure/privilege protection level of the write transaction controlling a command write sequence or aborting a command operation. Writes to load an existing FCCOB register, clear CCIF, or clear PERDY must occur with the same protection level as indicated by the CMDPRT bits. Writes to clear the CWSABT, ACCERR, PVIOL, or CMDABT flags are not influenced by the state of the CMDPRT bits. If FCTRL[ABTREQ] is set while CCIF is high or low, the CMDPRT bits will change to reflect the security/privilege protection level of the write transaction that set the ABTREQ bit. 00b - Secure, normal access 01b - Secure, privileged access 10b - Nonsecure, normal access 11b - Nonsecure, privileged access
7 CCIF	Command Complete Interrupt Flag The CCIF flag indicates that a flash command has completed. The CCIF flag is cleared by writing a 1 to CCIF to launch a flash command. The CCIF flag stays low until the command completes.

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	The CCIF flag is cleared by a mass erase request with CCIF set. The CCIF flag is also cleared upon entry into Deep Sleep, Power Down, and Deep Power Down modes and stays low until the end of recovery from these non-active power modes. The CCIF flag is reset to 0 but is set to 1 at the end of flash initialization. 0b - Flash command, initialization, or power mode recovery in progress 1b - Flash command, initialization, or power mode recovery has completed
6 CWSABT	Command Write Sequence Abort Flag The CWSABT flag indicates whether a request to abort a command write sequence prior to command launch has been granted. If FCTRL[ABTREQ] is set while CCIF is high, the CWSABT flag will get set. Once CWSABT is set, ABTREQ will get cleared. While the CWSABT flag is set, ABTREQ cannot be set. The CWSABT flag is cleared by writing a 1 to CWSABT while CCIF is set or clear and ABTREQ is clear. Writing a 0 to the CWSABT flag has no effect. The CWSABT flag is cleared by a mass erase request with CCIF set. 0b - Command write sequence not aborted 1b - Command write sequence aborted
5 ACCERR	Command Access Error Flag The ACCERR flag indicates an illegal attempt was made to launch a flash command due to a violation of the command write sequence or by providing invalid command parameters. The ACCERR flag also sets during command execution if FMU parameters were not loaded from IFR1 during initialization. While the ACCERR flag is set, the CCIF flag cannot be cleared to launch a command. The ACCERR flag is cleared by writing a 1 to ACCERR while CCIF is set. Writing a 0 to the ACCERR flag has no effect. The ACCERR flag is cleared by a mass erase request with CCIF set but is not cleared during Power Down recovery. 0b - No access error detected 1b - Access error detected
4 PVIOL	Command Protection Violation Flag The PVIOL flag indicates an attempt was made to modify a protected area of flash memory during a command operation. While the PVIOL flag is set, the CCIF flag cannot be cleared to launch a command. The PVIOL flag is cleared by writing a 1 to PVIOL while CCIF is set. Writing a 0 to the PVIOL flag has no effect. The PVIOL flag is cleared by a mass erase request with CCIF set but is not cleared during Power Down recovery. 0b - No protection violation detected 1b - Protection violation detected
3 —	Reserved
2	Command Abort Flag

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
CMDABT	The CMDABT flag indicates FCTRL[ABTREQ] was set during a command operation while CCIF was clear. This event will result in the termination of the operation unless it occurs during the exit routine of the operation. Once CMDABT is set, ABTREQ will get cleared. While the CMDABT flag is set, ABTREQ cannot be set and the CCIF flag cannot be cleared to launch a command. The CMDABT flag is cleared by writing a 1 to CMDABT while CCIF is set and ABTREQ is clear. Writing a 0 to the CMDABT flag has no effect. The CMDABT flag is cleared by a mass erase request with CCIF set. 0b - No command abort detected 1b - Command abort detected
1 —	Reserved
0 FAIL	Command Fail Flag The FAIL flag is set if an error is detected during execution of a flash command, mass erase operation, or flash initialization. If the FAIL flag sets along with ACCERR after launching a flash command or mass erase operation, FMU parameters were not loaded from IFR1 during initialization and a clean POR where FMU parameters are successfully loaded is required to clear the FAIL flag. As a status flag, this bit cannot (and need not) be cleared by the user like some of the other flags in this register. The value of the FAIL flag for "command-N" is valid only at the end of the "command-N" execution when CCIF=1 and before the next command has been launched. At some point during the execution of "command-N+1," the FAIL flag is cleared unless FMU parameters were not loaded from IFR1 during initialization. The FAIL flag is cleared by a mass erase request with CCIF set unless FMU parameters were not loaded from IFR1 during initialization. The FAIL flag is not cleared during Power Down recovery. 0b - Error not detected 1b - Error detected

5.6.2.1.3 Flash Configuration Register (FCNFG)

Offset

Register	Offset
FCNFG	4h

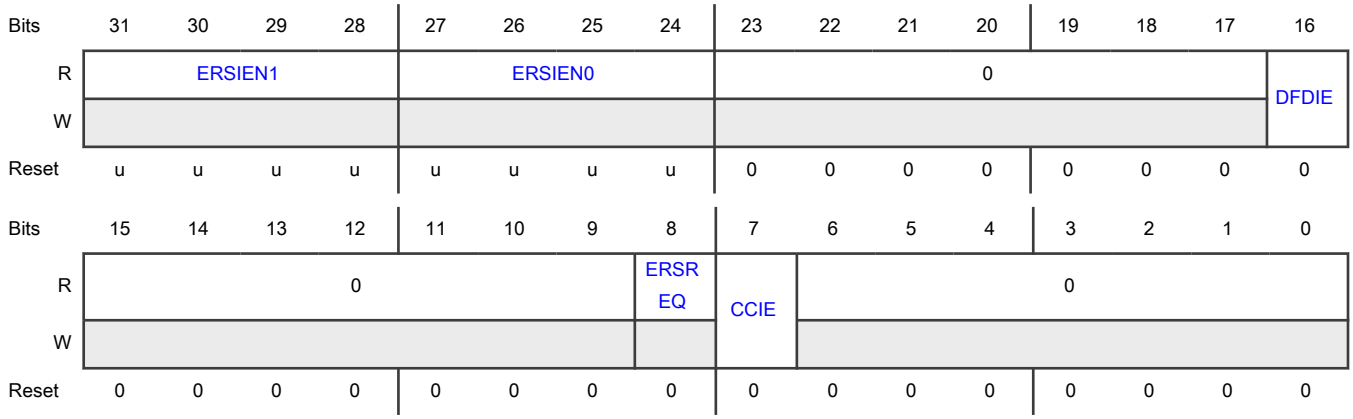
Function

This register provides information on the current functional state of the flash module.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram



Fields

Field	Function
31-28 ERSIEN1	Erase IFR Sector Enable - Block 1 (for dual block configs) This field controls the ability to erase the IFR sectors in block 1 using the ERSSCR command. These bits are loaded from sideband signals during initialization and at launch of ERSSCR command. Note: These bits have no effect on single block configurations. Note: The ERSALL command and MCU mass erase request do not adhere to these restrictions. X equals 0, 1, 2, or 3. Any and all bits in the field are independently configured. 0000b - Block 1 IFR Sector X is protected from erase by ERSSCR command 0001b - Block 1 IFR Sector X is not protected from erase by ERSSCR command
27-24 ERSIEN0	Erase IFR Sector Enable - Block 0 This field controls the ability to erase the IFR sectors in block 0 using the ERSSCR command. These bits are loaded from sideband signals during initialization and at launch of ERSSCR command. Note: The ERSALL command and MCU mass erase request do not adhere to these restrictions. X equals 0, 1, 2, or 3. Any and all bits in the field are independently configured. 0000b - Block 0 IFR Sector X is protected from erase by ERSSCR command 0001b - Block 0 IFR Sector X is not protected from erase by ERSSCR command
23-17 —	Reserved
16 DFDIE	Double Bit Fault Detect Interrupt Enable The DFDIE bit controls interrupt generation when an uncorrectable ECC fault is detected during a valid flash read access from the platform flash controller. 0b - Double bit fault detect interrupt disabled 1b - Double bit fault detect interrupt enabled. An interrupt request is generated whenever the FSTAT[DFDIF] flag is set.

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
15-9 —	Reserved
8 ERSREQ	Mass Erase Request This bit indicates whether a sideband request has been received by the command controller to execute the Mass Erase operation. ERSREQ is not directly writable but sets when a Mass Erase request is received by the flash module while CCIF is set. ERSREQ is cleared by the command controller when the operation completes. 0b - No request or request complete 1b - Request to run the Mass Erase operation
7 CCIE	Command Complete Interrupt Enable The CCIE bit controls interrupt generation when a flash command completes. 0b - Command complete interrupt disabled 1b - Command complete interrupt enabled. An interrupt request is generated whenever the FSTAT[CCIF] flag is set.
6-0 —	Reserved

5.6.2.1.4 Flash Control Register (FCTRL)

Offset

Register	Offset
FCTRL	8h

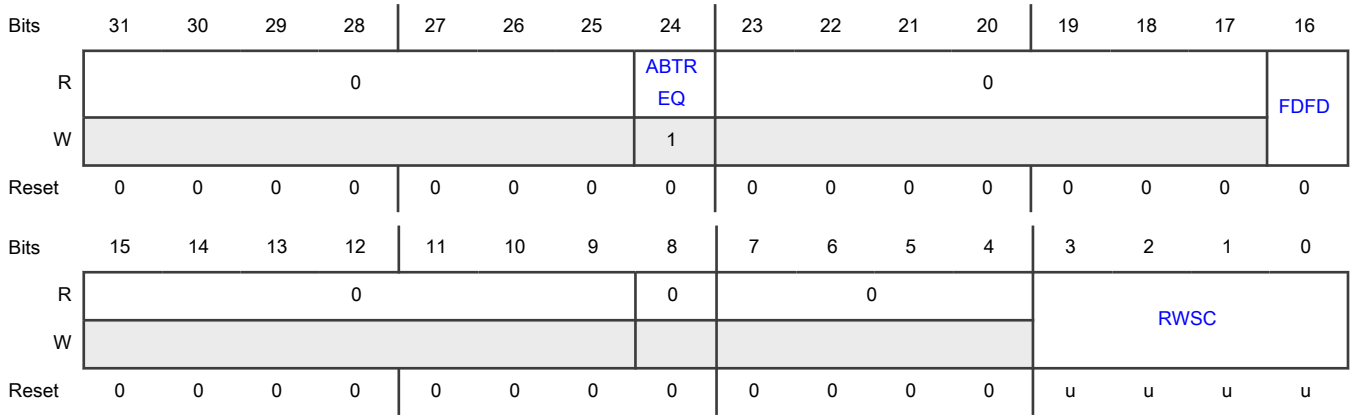
Function

The flash control register controls activity associated with flash memory reads and command operations.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram



Fields

Field	Function
31-25 —	Reserved
24 ABTREQ	Abort Request With FSTAT[CMDP] set, setting the ABTREQ bit allows a user to abort a command write sequence or command operation. The ABTREQ bit can only be set by a user with the same domain ID as indicated by FSTAT[CMDDID] and the same or higher security/privilege protection level than is indicated by FSTAT[CMDPRT]. While ABTREQ is set, CCIF cannot be cleared to launch a command. If FSTAT[CCIF] is high when ABTREQ is set, the command write sequence will abort and ownership transferred to the user setting the ABTREQ bit. The FSTAT[CWSABT] flag will set and the CMDPRT field will be updated based on the write transaction that sets the ABTREQ bit. The ABTREQ bit will then be cleared automatically. If FSTAT[CCIF] is low when ABTREQ is set, the command operation will abort and ownership transferred to the user setting the ABTREQ bit. The FSTAT[CMDABT] flag will set and the CMDPRT field will be updated based on the write transaction that sets the ABTREQ bit. The ABTREQ bit will then be cleared automatically. While CWSABT or CMDABT are high, writes to ABTREQ will be ignored. 0b - No request to abort a command write sequence 1b - Request to abort a command write sequence
23-17 —	Reserved
16 FDFD	Force Double Bit Fault Detect The FDFD bit enables the user to emulate the setting of the FSTAT[DFDIF] flag to check the associated interrupt routine. The FDFD bit is cleared by writing a 0 to FDFD. 0b - FSTAT[DFDIF] sets only if a double bit fault is detected during a valid flash read access from the platform flash controller

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	1b - FSTAT[DFDIF] sets during any valid flash read access from the platform flash controller. An interrupt request is generated if the DFDIE bit is set.
15-9 —	Reserved
8 —	Reserved
7-4 —	Reserved
3-0 RWSC	Read Wait-State Control These bits control the number of wait-states added to account for the ratio of system clock period to flash access time during full speed power mode. Ratios greater than one require non-zero settings for the RWSC field for proper flash accesses. The required settings are documented in the device data sheet. 0h - no additional wait-states are added (single cycle access) 1h - 1 additional wait-state is added 2h - 2 additional wait-states are added ... Fh - 15 additional wait-states are added These bits are loaded from IFR1 during initialization. **NOTE** The default FCTRL[RWSC] is set to a value appropriate for the chip's max operating frequency. Refer to the chip-specific section for the default value.

5.6.2.1.5 Flash Common Command Object Registers (FCCOB0 - FCCOB7)

Offset

For a = 0 to 7:

Register	Offset
FCCOBa	10h + (a × 4h)

Function

The FCCOB register group provides fields for command codes and parameters. The individual words within the set append a 0-7 hex identifier to the FCCOB register name: FCCOB0, FCCOB1, ..., FCCOB7.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	CCOBn															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CCOBn															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-0 CCOBn	CCOBn The FCCOB register group provides a command code and relevant parameters to the command controller. The individual registers that compose the FCCOB data set can be written in any order, but you must provide all needed values, which vary from command to command. First, set up all required FCCOB fields and then initiate the command execution by writing a 1 to the FSTAT[CCIF] bit. This clears the CCIF bit, which locks all FCCOB parameter fields and they cannot be changed by the user until the command completes (CCIF returns to 1). No command buffering or queueing is provided; the next command can be loaded only after the current command completes. Some commands return information to the FCCOB registers. Any values returned to FCCOB are available for reading after the FSTAT[CCIF] flag returns to 1. The following table shows a generic flash command format. The first FCCOB register, FCCOB0, always contains the command code. This 8-bit value defines the command to be executed. The command code is followed by the parameters required for this specific flash command, typically an address and/or data values.

NOTE

The command parameter table is written in terms of FCCOB Number (which is equivalent to the word number). This number is a reference to the FCCOB register name and is not the register address.

FCCOB Number[1]	Typical Command Parameter Contents [31:0]
0	FCMD [7:0] (flash command code, bits [31:8] are not writable)
1	FCMDOPT [7:0] (flash command options, bits [31:8] are not writable)
2	Flash address 1 (start)
3	Flash address 2 (end)
4	Data Word 0
5	Data Word 1
6	Data Word 2

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Field	Function	
	FCCOB Number[1]	Typical Command Parameter Contents [31:0]
	7	Data Word 3
	1. Refers to FCCOB register name, not register address	

5.7 Glossary

Block Checker — MCU IP that provides a mechanism to set access rights to the flash memory for read and write operations.

Command write sequence — A series of MCU writes to the Flash FCCOB register group and FSTAT[CCIF] that initiates and controls the execution of flash algorithms that are built into the flash module.

Endurance — The number of times that an aligned flash or IFR phrase can be erased and reprogrammed.

FCCOB (Flash Common Command Object Block) — A group of flash registers that are used to pass command, address, data, and any associated parameters to the command controller.

Flash block — A macro within the flash module which provides the nonvolatile memory storage with an aligned block address being the first address in the block.

Flash module — All flash blocks plus a flash management unit providing command control and an interface to MCU buses.

Flash page — 128 bytes of flash main memory with an aligned page having byte-address[6:0] = 00h; represents the largest portion of the flash memory that can be programmed in one operation.

Flash phrase — 16 bytes of flash main memory with an aligned phrase having byte-address[3:0] = 0000b; represents the smallest portion of the flash memory that can be programmed in one operation.

Flash sector — 8 Kbytes of flash main memory with an aligned sector having byte-address[12:0] = 0000h; represents the smallest portion of the flash memory that can be erased in one operation.

FLW — The Flash Logical Window module allows a specific logical address range to access a programmable physical address range in the flash memory with programmable FLW default settings stored in IFR space.

FMC — Flash Memory Controller module manages flash memory reads and writes.

IFR — Information flash region separate from the main flash memory array. Each flash block has 32 Kbytes of user IFR space. Sometimes referred to as IFR0.

IFR1 — Information flash region separate from the main memory array reserved for array trim, MCU trim, and test. Each flash array has 8 Kbytes of IFR1 space.

IFR page — 128 bytes of IFR or IFR1 space with an aligned IFR page having byte-address[6:0] = 00h; represents the largest portion of IFR space that can be programmed.

IFR phrase — 16 bytes of IFR or IFR1 space with an aligned IFR phrase having byte-address[3:0] = 0000b; represents the smallest portion of IFR space that can be programmed in one operation.

IFR sector — 8 Kbytes of IFR or IFR1 space with an aligned IFR sector having byte-address[12:0] = 0000h; represents the smallest portion IFR space that can be erased.

MISR — Multiple-input signature register used to generate a signature based on flash memory contents read.

MSF — Microcontroller Secure Flash.

NVM — Nonvolatile memory. The flash block is an NVM using NOR-type flash memory technology.

Program flash — Program flash memory provides nonvolatile storage for vectors and code store.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Retention — The length of time that data (erased or programmed) can be kept in the NVM without experiencing errors upon readout.

Security/Privilege protection levels — The Flash module supports 4 levels of security/privilege protection for flash commands:

1. Secure (Trusted) / Privileged
2. Secure (Trusted) / User
3. Non-secure (Non-Trusted) / Privileged
4. Non-secure (Non-Trusted) / User

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Chapter 6
Flash Memory Controller (FMC)

6.1 Chip-specific FMC information

Table 58. Reference links to related information

Topic	Related module	Reference
Full description	FMC	FMC
System memory map		Memory map
Clocking		Clock distribution
Power Management		Power Management
Signal multiplexing	Port control	Signal multiplexing

See NVM Control (NVM_CTRL) for details on flash speculation.

6.1.1 Module instances

This chip has one instance of the FMC module, FMC0.

6.2 Overview

The Flash Memory Controller (FMC) is a memory interface and acceleration unit providing:

- An interface between the device and the nonvolatile memory
- Buffers that can accelerate flash memory transfers
  - A flash-phrase-sized(128 bits) single-entry buffer holds the most recently accessed flash phrase.
  - An optional flash-phrase-sized(128 bits) speculation buffer can prefetch the next flash phrase.

The FMC manages the interface between the device and the flash memory. The FMC receives status information describing the configuration of the memory and uses this information to ensure a proper interface. The following table shows the supported read and write operations.

Flash memory type	Read	Write
Program, IFR0, IFR1 flash memory	8-bit, 16-bit, and 32-bit reads	16-byte and 128-byte writes

The FMC is controlled by a programmer's model external to the FMC module; see the chip-specific FMC information for details. The FMC's programming model provides a very configurable, high performance flexible memory controller, which can be optimized for the runtime characteristics of specific applications.

NOTE

Program the FMC's controls only while the flash controller is idle. Changing the configuration settings while a flash access is in progress can cause non-deterministic, unpredictable behavior.

6.2.1 Features

- Interface between the device and the flash memory:
  - The FMC's input bus supports 8-bit, 16-bit, and 32-bit read operations to flash memory.
  - The FMC's flash memory interface fetches a 128-bit flash phrase.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

- For input read requests, the FMC fetches a flash phrase with the desired read data from flash memory.
- The flash memory interface can write aligned 16-byte flash write phrases or aligned 128-byte flash write pages to flash memory.
- The FMC has a 16-byte aligned write buffer. This buffer is used once for aligned 16-byte flash write phrases or eight times for aligned 128-byte flash write pages.
- The FMC's input bus supports 32-bit write operations for flash memory writes to fill the FMC's write buffer.
- For input write requests, the FMC must receive the 4-word(16-byte) write of an aligned phrase in order.
- Program flash and IFR0 support read and write requests, IFR1 only supports read requests.

- Acceleration of data transfer from flash memory to the device:
  - A flash-phrase-sized single-entry buffer that holds the current flash phrase fetched due to a FMC read request. Subsequent FMC read requests *that hit in the single-entry buffer* return data with no wait states.
  - A flash-phrase-sized prefetch speculation buffer with controls for prefetching on instructions and/or data reads. When prefetching is enabled, idle FMC-to-flash interface cycles are used to fetch the next sequential flash phrase and hold it in the prefetch buffer. Subsequent FMC read requests *that hit in the speculation buffer* return data with no wait states.
  - Input controls:
    - To disable data type speculation
    - To disable all speculation
    - To clear the single-entry buffer and speculation buffer
    - To disable the single-entry buffer and speculation buffer
    - To enable AHB bus stall on flash busy
    - To lock access to IFR1
    - To disable AHB bus error on flash multibit ECC error for data
    - To disable AHB bus error on flash multibit ECC error for instruction

    To enable flash speculation, enable AHB bus error on flash multibit ECC error for data and instruction. See the chip-specific section for details about controls.

NOTE

Clear the speculation buffer before accessing recently modified flash addresses. To clear the speculation buffer, first disable then re-enable the speculation via other modules.

6.3 Functional description

The FMC is a flash interface and acceleration unit, with flexible buffers for user configuration.

- The FMC's input bus can operate faster than the flash memory.
- The FMC-to-flash interface has flow control to add wait states as needed (for input bus reads that need flash accesses).
- The FMC also contains various configurable buffers that hold recent flash accesses. If an input bus read hits a valid buffer, then that access will complete with no wait states.

6.3.1 Modes of operation

The FMC only operates when a bus master accesses the flash memory.

For any device power mode where the flash memory cannot be accessed, FMC is disabled automatically.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

6.3.2 Default configuration

After system reset, the FMC is configured to provide a significant level of buffering for transfers from the flash memory. For all banks:

- The single-entry and speculation buffers are cleared by reset.
- Prefetch support by speculation buffer for data and instructions is enabled.

6.3.3 Configuration options

The default configuration provides a high degree of flash acceleration, but advanced users may want to customize the FMC buffer configurations to maximize throughput for their use cases. When reconfiguring the FMC for custom use cases, do not program the FMC's control registers while the flash memory is being accessed. Instead, change the control registers with a routine executing from RAM in Supervisor mode.

The FMC's buffering controls allow other modules to tune resources to suit specific application requirements. The buffer are each controlled individually. The controls enable caching and prefetching per access type (instruction fetch or data reference).

See the chip-specific section for details about the registers used to configure FMC.

As an application example: if both instruction fetches and data references are accessing flash memory, then control is available to send instruction fetches, data references, or both to the single-entry buffer. Likewise, speculation can be enabled or disabled for either type of access.

6.3.4 Wait states

Because the core, crossbar switch, and bus masters can be clocked at a higher frequency than the flash clock, flash memory accesses that do not hit in the speculation buffer usually require wait states.

FMC does not allow the configuration of wait states directly. Wait states can be controlled via FMU FCTRL[RWSC].

6.3.5 Speculative reads

The FMC has a speculation buffer that reads ahead to the next phrase in the flash memory if there is an idle cycle. Speculative prefetching is programmable for instruction and data accesses. Because many code accesses are sequential, using the speculative prefetch buffer improves performance in most cases.

Program flash and IFR0 support speculation reads, IFR1 doesn't support speculation reads. See the chip-specific section for information about controlling speculative reads.

When speculative reads are enabled, the FMC immediately requests the next sequential phrase address after a read completes. By requesting the next phrase immediately, speculative reads can help to reduce or even eliminate wait states when accessing sequential code and/or data.

6.3.6 Interrupts

This module has no interrupts.

6.4 External signals

The FMC has no external signals.

6.5 Initialization and application information

The FMC does not require user initialization. Flash acceleration features are enabled by default.

The FMC has no visibility into flash memory erase and program cycles because the Flash Memory module manages them directly. To prevent the possibility of returning stale data after an application executes flash memory commands, disable and/or clear FMC's single-entry buffer and speculation buffer. If you want to disable the speculation buffer and cache in certain specific cases, we recommend disabling them all.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

See the chip-specific section for details about the registers used to disable and/or clear speculation buffer and single-entry buffer.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Chapter 7
Error Injection Module (EIM)

7.1 Chip-specific EIM information

Table 59. Reference links to related information

Topic	Related module	Reference
Full description	EIM	Error Injection
System memory map		Memory map
Clocking		Clock distribution
Power Management		Power Management
Signal multiplexing	Port control	Signal multiplexing

7.1.1 Module instances

This chip contains one instance of the Error Injection module, EIM0.

7.1.2 EIM channel mapping

This chip provides EIM module for ECC RAM. It can inject error when reading ECC RAM. It can be used for self-test. In this chip, EIM has one error injection channel to recode SRAM A0.

7.1.3 Access modes

- If the privilege access is disabled (SYSCON.PROTLVL[PRIV] = 0b), this peripheral is accessible in the user mode.
- If the privilege access is enabled (SYSCON.PROTLVL[PRIV] = 1b), this peripheral is accessible only in the privilege mode.

7.2 Overview

The Error Injection Module (EIM) is mainly used for diagnostic purposes. It provides a method to test the diagnostics (memory ECC, interconnect parity) by error injection in the field. See the chip-specific EIM information to determine which functional safety features are supported by this method.

EIM enables you to inject artificial errors on error-checking mechanisms of a system, such as ECC for RAM read data and parity bits. For each such mechanism that EIM supports on the chip, EIM can inject single-bit and multi-bit inversions on data in the applicable target bus. Injecting faults on memory accesses can be used to exercise the SEC-DED ECC function of the related system.

NOTE

Terminology in this chapter has been updated as follows:

Table 60. Updated terms

Updated term	Deprecated term
Controller	Master

7.2.1 Features

The EIM includes these features:

- Supports upto 32 error injection channel. See the chip-specific EIM information for channel assignment details.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

- Protection against accidental enable and reconfiguration error injection function via two-stage enable mechanism

7.2.2 Block diagram

The following diagram shows an example of EIM implementation with a 64-bit read data bus and an 8-bit checkbit bus.

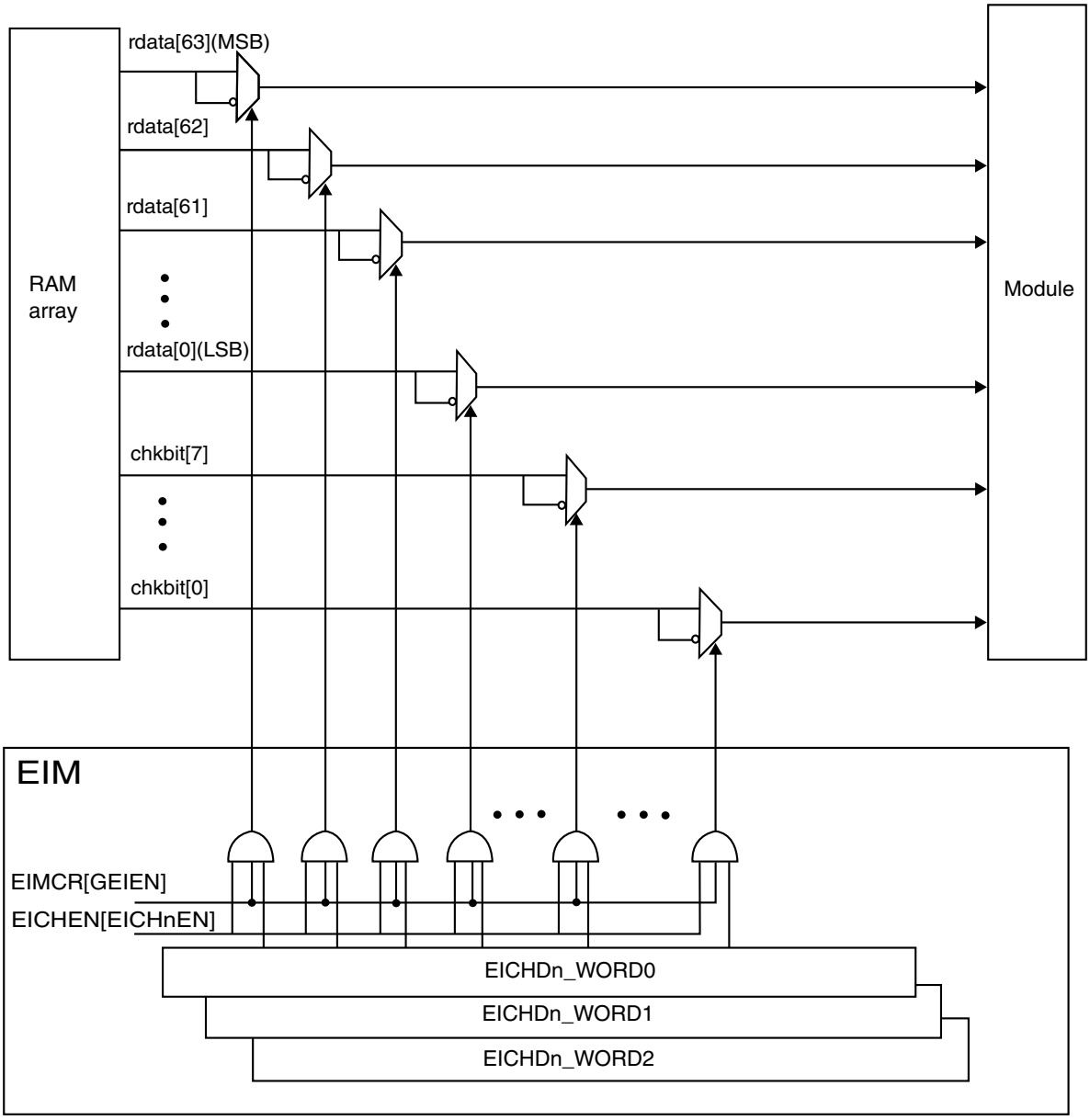


Figure 10. EIM functional block diagram (64-bit read data bus and 8-bit check bit bus)

Several memory elements are implemented within a device, which may not only be the large memory blocks (Flash and SRAM) but also smaller memories like caches, the TCD blocks, and the embedded peripheral memories. Some larger memories may actually be built from multiple memory elements, dependent on their size or function. Each of these memory elements implements its own control logic, the memory controller, that performs the accesses to the actual memory, the memory array. An EIM channel is associated with a memory controller and provides the capability to alter one or multiple signals in the read access path from the corresponding memory array(s). Only memory controllers controlling a safety related memory may be associated with an EIM channel.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

7.3 Functional description

The EIM provides protection against accidental enabling and reconfiguration of the error injection function by enforcing a two-stage enablement mechanism. To properly enable the error injection mechanism for a channel:

- Write 1 to the EICHEN[EICH*n*EN] field, where *n* denotes the channel number.
- Write 1 to EIMCR[GEIEN].

NOTE

When the use case for a channel requires writing any EICHD*n*_WORD register, write the EICHD*n*_WORD register before executing the two-stage enablement mechanism. A successful write to any EICHD*n*_WORD register clears the corresponding EICHEN[EICH*n*EN] field.

See the chip-specific EIM information for channel assignment details. Each channel:

- Can be assigned to a single memory array interface by intercepting the assigned memory read data bus and checkbit bus, and injects errors by inverting the value transmitted for selected bits on each bus line.
- Can be assigned to a redundant comparison unit by intercepting the signals being compared, and injecting errors by inverting the value transmitted for selected bits on each bus line.

On a memory read access, the applicable EICHD*n*_WORD registers define which bits of the read data and/or checkbit bus to invert.

Figure 10 depicts the interception and override of a 64-bit read data bus and an 8-bit checkbit data bus for an example memory array.

Error injection scenarios

The EIM supports these cases of error injection:

- To generate a single-bit error, invert only 1 bit of the CHKBIT_MASK or DATA_MASK in the EICHD*n*_WORD registers.
- To generate a multi-bit error, invert only 2 bits of the CHKBIT_MASK or DATA_MASK in the EICHD*n*_WORD registers.

NOTE

An attempt to invert more than 2 bits in one operation might result in undefined behavior.

To enable error injection:

1. Set the EICHD*n*_WORD*m*[CHKBIT_MASK] and EICHD*n*_WORD*m*[B*a*_*b*DATA_MASK] fields for each channel that will be driving an injection.
2. Program the EICHEN register to enable the channels that will be injecting errors.
3. Set the EIMCR[GEIEN] field to globally allow all enabled channels to actively inject errors.

To disable error injection, either disable the EIMCR[GEIEN] field or disable the individual channel enable fields of the EICHEN register.

7.4 Initialization

This module does not require initialization.

7.5 EIM register descriptions

The EIM provides a programming model mapped to an on-platform peripheral slot.

Programming model access

All system bus controllers can access the programming model:

- Only in supervisor mode

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

- Using only 32-bit (word) accesses

Any of the following attempted references to the programming model generates an error termination:

- In user mode
- Using non-32-bit access sizes
- To undefined (reserved) addresses

Attempted updates to the programming model while the EIM is in the midst of an operation result in non-deterministic behavior.

Error injection channel descriptor: function and structure

Each error injection channel descriptor:

- Specifies a mask that defines which bits of the read data and/or checkbit bus from target RAM are inverted on a read access.
- Consists of a 128-bit (16-byte) structure, composed of four 32-bit words, in the EIM programming model. Unused words are not documented.
  - Word0 (EICHD*n*_WORD0), if present, defines the checkbit mask.
  - Word1-3 (EICHD*n*_WORD1-3), if present, define the data mask. Word2 and Word3 are present only when required by the total width of the channel's data mask. See Error injection channel descriptor: DATA_MASK details.

Error injection channel descriptor: DATA_MASK details

The following table shows the total width of DATA_MASK and the distribution of its bits across the WORD registers.

Table 61. Error injection channel descriptor: DATA_MASK details

Channel	DATA_MASK total width (bits)	Specific bits of DATA_MASK in		
		WORD1	WORD2	WORD3
0	32	31-0	—	—

7.5.1 EIM memory map

EIM0 base address: 4008_C000h

Offset	Register	Width (In bits)	Access	Reset value
0h	Error Injection Module Configuration Register (EIMCR)	32	RW	0000_0000h
4h	Error Injection Channel Enable register (EICHEN)	32	RW	0000_0000h
100h	Error Injection Channel Descriptor 0, Word0 (EICHD0_WORD0)	32	RW	0000_0000h
104h	Error Injection Channel Descriptor 0, Word1 (EICHD0_WORD1)	32	RW	0000_0000h

7.5.2 Error Injection Module Configuration Register (EIMCR)

Offset

Register	Offset
EIMCR	0h

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Function

The EIM Configuration Register is used to globally enable/disable the error injection function.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															GEIEN
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-1 —	Reserved
0 GEIEN	Global Error Injection Enable This bit globally enables or disables the error injection function of the EIM. This field is initialized by hardware reset. 0b - Disabled 1b - Enabled

7.5.3 Error Injection Channel Enable register (EICHEN)

Offset

Register	Offset
EICHEN	4h

Function

Each field of the Error Injection Channel Enable register (EICHEN) is used to enable or disable the corresponding error injection channel.

NOTE

To enable an error injection channel, the Global Error Injection Enable (EIMCR[GEIEN]) field must also be asserted.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	EICH0EN	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31 EICH0EN	Error Injection Channel 0 Enable This field enables the corresponding error injection channel. The Global Error Injection Enable (EIMCR[GEIEN]) field must also be asserted to enable error injection. After error injection is enabled, all subsequent read accesses incur one or more bit inversions as defined in the corresponding EICHD*n*_WORD registers. Error injection remains in effect until the error injection channel is manually disabled via software. Any write to the corresponding EICHD*n*_WORD registers clears the corresponding EICHEN[EICH*n*EN] field, disabling the error injection channel. 0b - Error injection is disabled on Error Injection Channel 0 1b - Error injection is enabled on Error Injection Channel 0
30 —	Reserved
29 —	Reserved
28 —	Reserved
27 —	Reserved
26 —	Reserved
25 —	Reserved

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
24 —	Reserved
23 —	Reserved
22 —	Reserved
21 —	Reserved
20 —	Reserved
19 —	Reserved
18 —	Reserved
17 —	Reserved
16 —	Reserved
15 —	Reserved
14 —	Reserved
13 —	Reserved
12 —	Reserved
11 —	Reserved
10	Reserved

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
—	
9 —	Reserved
8 —	Reserved
7 —	Reserved
6 —	Reserved
5 —	Reserved
4 —	Reserved
3 —	Reserved
2 —	Reserved
1 —	Reserved
0 —	Reserved

7.5.4 Error Injection Channel Descriptor 0, Word0 (EICHD0_WORD0)

Offset

Register	Offset
EICHD0_WORD0	100h

Function

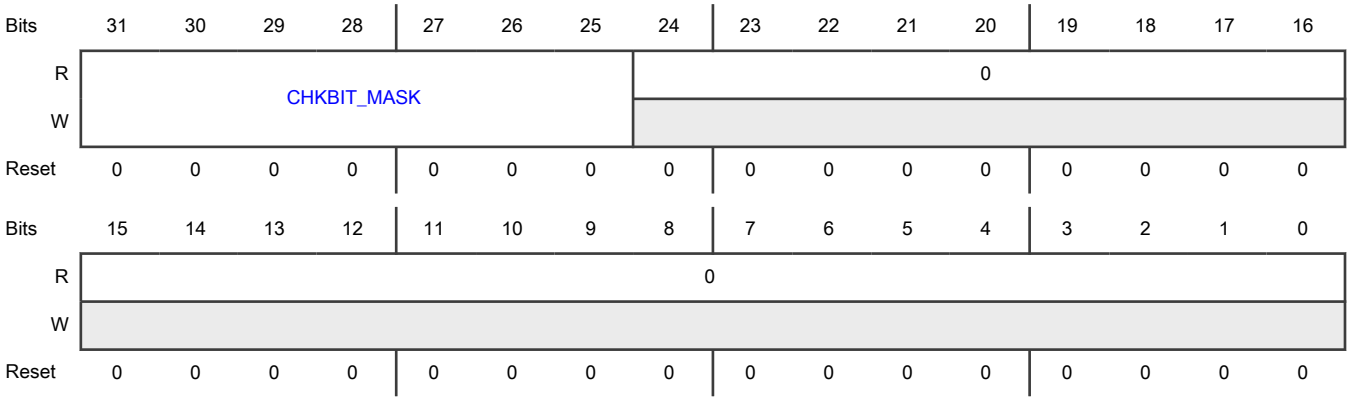
The first word of the Error Injection Channel Descriptor defines a left-justified mask field: CHKBIT_MASK. Each bit of CHKBIT_MASK specifies whether the corresponding bit of the checkbit bus from the target RAM should be inverted or

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

remain unmodified on read accesses. Successful write to this field clears the corresponding error injection channel valid bit, EICHEN[EICH*n*EN].

Diagram



Fields

Field	Function
31-25 CHKBIT_MASK	Checkbit Mask This field defines a bit-mapped mask that specifies whether the corresponding bit of the checkbit bus from the target RAM should be inverted or remain unmodified. Writes to unimplemented bits are ignored. For any unique details about the mapping of CHKBIT_MASK's bits to a channel's target RAM, see the chip-specific EIM information. **NOTE** Because CHKBIT_MASK is left-justified, the highest bit in the bit range is always in the position of the most significant bit. For CHKBIT_MASK[6:0] (7 bits wide), CHKBIT_MASK[6] is in the position of the most significant bit. 0b - The corresponding bit of the checkbit bus remains unmodified. 1b - The corresponding bit of the checkbit bus is inverted.
24-0 —	Reserved

7.5.5 Error Injection Channel Descriptor 0, Word1 (EICHD0_WORD1)

Offset

Register	Offset
EICHD0_WORD1	104h

Function

The second word of the Error Injection Channel Descriptor defines a right-justified mask field. The bits in B0_3DATA_MASK correspond to EICHD0_WORD1 of the target bus. Each bit specifies whether the corresponding bit of the target bus should be

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

inverted or remain unmodified on read accesses. A successful write to this field clears the corresponding error injection channel valid field, EICHEN[EICH*n*EN].

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	B0_3DATA_MASK															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	B0_3DATA_MASK															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-0 B0_3DATA_MASK	Data Mask Field This field defines a bit-mapped mask that specifies whether the corresponding bit of the read data bus from the target RAM should be inverted or remain unmodified. Writes to unimplemented bits are ignored. **NOTE** For the specific DATA_MASK bits to which B0_3DATA_MASK corresponds, See Error injection channel descriptor: DATA_MASK details. 0b - The corresponding bit of EICHD0_WORD1 on the read data bus remains unmodified. 1b - The corresponding bit of EICHD0_WORD1 on the read data bus is inverted.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Chapter 8
Error Recording Module (ERM)

8.1 Chip-specific Error Recording information

Table 62. Reference links to related information

Topic	Related module	Reference
Full description	ERM	Error Recording
System memory map		Memory map
Clocking		Clock distribution
Power Management		Power Management
Signal multiplexing	Port control	Signal multiplexing

8.1.1 Module instances

This chip contains one instance of the Error Recording module, ERM0.

8.1.2 ERM channel mapping

ERM can record the address and ECC check bits (errsyndrome), when ECC error happens. ERM can be configured to generate interrupt when single bit or multi bit error happens.

Table 63. ERM channel mapping

Channel no.	Module	Captured status
0	RAM A0	Single-bit error, multiple bit error, error address syndrome
1	FMU0	ECC double-bit error detected on read data from array to the FMC

8.1.3 Access modes

- If the privilege access is disabled (SYSCON.PROTLVL[PRIV] = 0b), this peripheral is accessible in the user mode.
- If the privilege access is enabled (SYSCON.PROTLVL[PRIV] = 1b), this peripheral is accessible only in the privilege mode.

8.2 Overview

The Error Reporting Module (ERM) provides information and optional interrupt notification on memory error events associated with ECC and parity. The ERM collects error events on memory accesses for memory arrays, such as flash memory, system RAM, or peripheral RAMs. ERM supports various channels for memory sources where each ERM channel is associated with a different memory module. See the chip-specific ERM information for details about supported memory sources and specific memory channel assignments. If the memory supports ECC, then ERM syndrome and error address information is captured along with the error event. ERM does not capture syndrome or error address for cache memories or memory with parity instead of ECC.

8.2.1 Features

The ERM includes these features:

All information provided in this document is subject to legal disclaimers.
© 2026 NXP B.V. All rights reserved.

- Optional interrupt notification on captured error events
- Capturing of address and syndrome information on single-bit correction and non-correctable ECC events
- Support for error event capturing for memory sources, with individual reporting fields and interrupt configuration per memory channel
- Recording the count value of the number of corrected error events

8.2.2 Block diagram

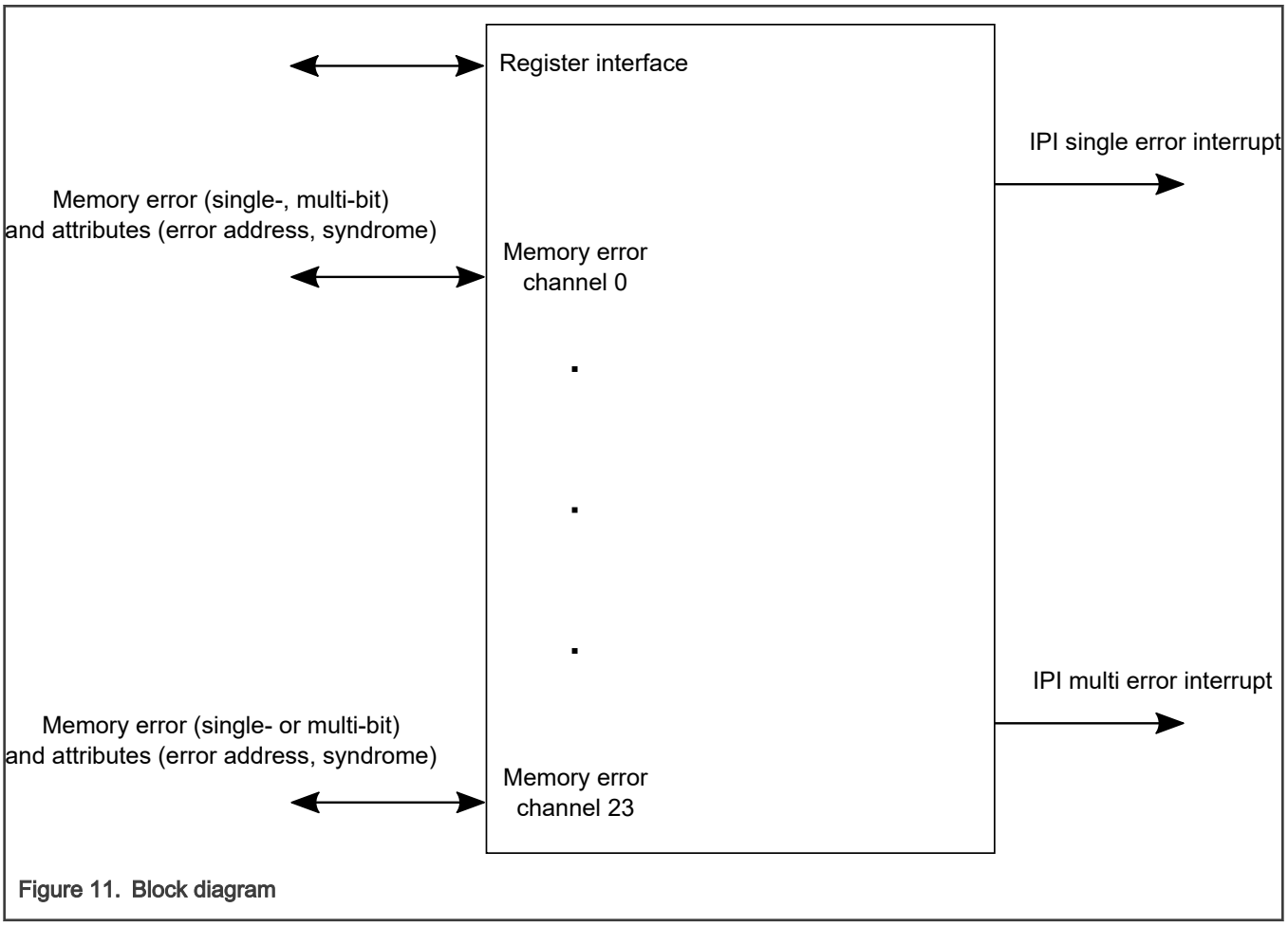


Figure 11. Block diagram

8.3 Functional description

8.3.1 Single-bit correction events

When a single-bit correction event on Memory *n* is detected, the ERM:

- Records the event by changing the value of the applicable Status Register bit SR*x*[SBC*n*] to 1.
- Increments the correctable error count value (until the counter reaches its maximum value): CORR_ERR_CNT*n*[COUNT].
- Records the corresponding access address that initiated the event in the Memory *n* Error Address Register: EAR*n* (if this register is present for the channel).
- Stores the corresponding ECC syndrome in the Memory *n* Error Syndrome Register: SYN*n* (if this register is present for the channel). This register identifies the bit position of the corrected data on single-bit data inversion.

The ERM holds event information only for the last reported event.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

To clear the record of an event, write 1 to SR*x*[SBC*n*] to change its value to 0.

To reset the correctable error count value, write all zeros to CORR_ERR_CNT*n*[COUNT].

Optional interrupt notification for single-bit correction events

The ERM provides an option to generate an interrupt notification upon the report of a single-bit correction event. To enable single-bit correction interrupts for a channel:

1. To enable interrupt notification for single-bit correction events on Memory *n*, set CR*x*[ESCIE*n*] to 1.
2. Subsequently, a single-bit correction event on Memory *n* is detected, the ERM:
   - Records the event and address, and stores the ECC syndrome as usual.
   - Additionally sends an interrupt notification corresponding to the event.
3. To clear both the record of an event and the corresponding interrupt notification, write 1 to SR*x*[SBC*n*] to change its value to 0.

8.3.2 Non-correctable error events

When a non-correctable ECC error event on Memory *n* is detected, the ERM:

- Records the event by changing the value of the applicable Status Register bit: SR*x*[NCE*n*] to 1.
- Records the corresponding access address that initiated the event in the Memory *n* Error Address Register: EAR*n* (if this register is present for the channel).
- Stores the corresponding ECC syndrome in the Memory *n* Error Syndrome Register: SYN*n* (if this register is present for the channel).
  - In the event of a non-correctable address bit inversion, SYN*n* identifies the pertinent address bit position.
  - In the event of a non-correctable, multi-bit data inversion, the syndrome value does not provide any additional diagnostic information.

The ERM holds event information only for the last reported event.

To clear the record of an event, write 1 to SR*x*[NCE*n*] to change its value to 0.

Optional interrupt notification for non-correctable error events

The ERM provides an option to generate an interrupt notification upon the report of a non-correctable ECC event. To enable non-correctable error interrupts for a channel:

1. To enable interrupt notifications for non-correctable error events on Memory *n*, set CR*x*[ENCIE*n*] to 1.
2. Subsequently, when a non-correctable error event on Memory *n* is detected, the ERM:
   - Records the event and address and stores the ECC syndrome as usual.
   - Additionally sends an interrupt notification corresponding to the event.
3. To clear both the record of an event and the corresponding interrupt notification, write 1 to SR*x*[NCE*n*] to change its value to 0.

NOTE

Parity errors can be mapped to non-correctable errors where error attributes like SYNDROME, ADDRESS are not provided.

8.4 Initialization

For each ERM channel supporting memory with ECC, prepare the corresponding memory array before enabling ERM interrupts about errors for that memory.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

1. Initialize the memory to a known value so that the correct corresponding ECC codeword is stored.
2. During the memory's initialization, if the ERM captures information about any ECC error event, clear the corresponding SR*x*[SBC*n*] or SR*x*[NCE*n*] field that stores the record of the event.
3. Program the applicable CR*x*[ESCIE*n*] and CR*x*[ENCIE*n*] fields to enable ERM interrupts as desired.

8.5 ERM register descriptions

You can access the programming model:

- Only in supervisor mode
- Using only 32-bit (word) accesses

Any of the following attempted references to the programming model generates an error termination:

- In user mode
- Using non-32-bit access sizes

Based on the design implementation, the following XFR error behavior is evident at the IPS interface.

- Within the ERM memory map, an XFR error is evident at reserved addresses from location 20h to FFh.
- No XFR error is evident at reserved addresses in memory spaces allocated to each channel. For example: For channel 0, for read/write accesses to reserved address 10Ch, the XFR error is 0.
- For accesses to locations beyond the addresses allocated for the final channel, the XFR error is 1.

NOTE

- See the chip-specific ERM information at the beginning of this chapter for details on Memory channel mapping.
- To access the channel registers, corresponding memory channel clock must be enabled.

8.5.1 ERM memory map

ERM0 base address: 4008_D000h

Offset	Register	Width (In bits)	Access	Reset value
0h	ERM Configuration Register 0 (CR0)	32	RW	0000_0000h
10h	ERM Status Register 0 (SR0)	32	RW	0000_0000h
100h	ERM Memory 0 Error Address Register (EAR0)	32	R	0000_0000h
104h	ERM Memory 0 Syndrome Register (SYN0)	32	R	0000_0000h
108h	ERM Memory 0 Correctable Error Count Register (CORR_ERR_CNT0)	32	RW	0000_0000h
118h	ERM Memory 1 Correctable Error Count Register (CORR_ERR_CNT1)	32	RW	0000_0000h

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

8.5.2 ERM Configuration Register 0 (CR0)

Offset

Register	Offset
CR0	0h

Function

This 32-bit control register configures the interrupt notification capability for available channels.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	ESCIE0	ENCIE0	0		ESCIE1	ENCIE1	0		Reserved		0		Reserved		0	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved		0		Reserved		0		Reserved		0		Reserved		0	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31 ESCIE0	ESCIE0 Enable Memory 0 Single Correction Interrupt Notification 0b - Interrupt notification of Memory 0 single-bit correction events is disabled. 1b - Interrupt notification of Memory 0 single-bit correction events is enabled.
30 ENCIE0	ENCIE0 Enable Memory 0 Non-Correctable Interrupt Notification 0b - Interrupt notification of Memory 0 non-correctable error events is disabled. 1b - Interrupt notification of Memory 0 non-correctable error events is enabled.
29-28 —	Reserved
27 ESCIE1	ESCIE1 Enable Memory 1 Single Correction Interrupt Notification 0b - Interrupt notification of Memory 1 single-bit correction events is disabled.

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	1b - Interrupt notification of Memory 1 single-bit correction events is enabled.
26 ENCIE1	ENCIE1 Enable Memory 1 Non-Correctable Interrupt Notification 0b - Interrupt notification of Memory 1 non-correctable error events is disabled. 1b - Interrupt notification of Memory 1 non-correctable error events is enabled.
25-24 —	Reserved
23-22 —	Reserved
21-20 —	Reserved
19-18 —	Reserved
17-16 —	Reserved
15-14 —	Reserved
13-12 —	Reserved
11-10 —	Reserved
9-8 —	Reserved
7-6 —	Reserved
5-4 —	Reserved
3-2 —	Reserved

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
1-0 —	Reserved

8.5.3 ERM Status Register 0 (SR0)

Offset

Register	Offset
SR0	10h

Function

This 32-bit status register reports error events for available channels.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	SBC0	NCE0	0		SBC1	NCE1	0		0				0			
W	W1C	W1C			W1C	W1C										
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0				0				0				0			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31 SBC0	SBC0 Memory 0 Single-Bit Correction Event Write 1 to clear this field.This write also clears the corresponding interrupt notification, if CR0[ESCIE0] is enabled. 0b - No single-bit correction event on Memory 0 detected. 1b - Single-bit correction event on Memory 0 detected.
30 NCE0	NCE0 Memory 0 Non-Correctable Error Event

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	Write 1 to clear this field.This write also clears the corresponding interrupt notification, if CR0[ENCIE0] is enabled. 0b - No non-correctable error event on Memory 0 detected. 1b - Non-correctable error event on Memory 0 detected.
29-28 —	Reserved
27 SBC1	SBC1 Memory 1 Single-Bit Correction Event Write 1 to clear this field.This write also clears the corresponding interrupt notification, if CR0[ESCIE1] is enabled. 0b - No single-bit correction event on Memory 1 detected. 1b - Single-bit correction event on Memory 1 detected.
26 NCE1	NCE1 Memory 1 Non-Correctable Error Event Write 1 to clear this field.This write also clears the corresponding interrupt notification, if CR0[ENCIE1] is enabled. 0b - No non-correctable error event on Memory 1 detected. 1b - Non-correctable error event on Memory 1 detected.
25-24 —	Reserved
23-20 —	Reserved
19-16 —	Reserved
15-12 —	Reserved
11-8 —	Reserved
7-4 —	Reserved
3-0	Reserved

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
—	

8.5.4 ERM Memory 0 Error Address Register (EAR0)

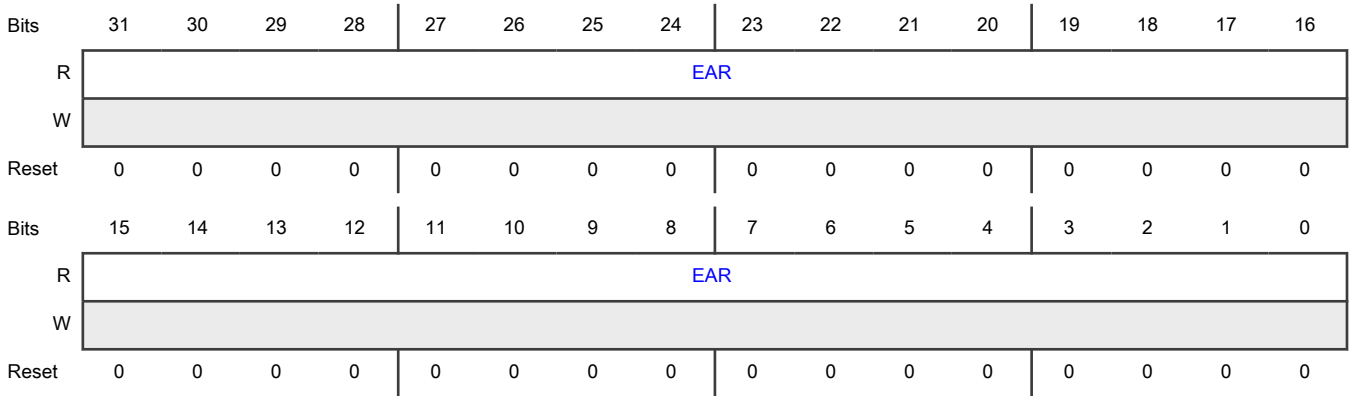
Offset

Register	Offset
EAR0	100h

Function

Each ERM Memory n Error Address Register is a 32-bit register for capturing the address of the last ECC event in Memory*n*, where*n* denotes the memory channel. Any attempted write to EAR*n* is ignored. See the chip-specific ERM information for the ERM Memory a Error Address Register.

Diagram



Fields

Field	Function
31-0 EAR	EAR Memory*n* Error Address — This field contains the faulting system address of the last recorded ECC event on Memory*n*.

8.5.5 ERM Memory 0 Syndrome Register (SYN0)

Offset

Register	Offset
SYN0	104h

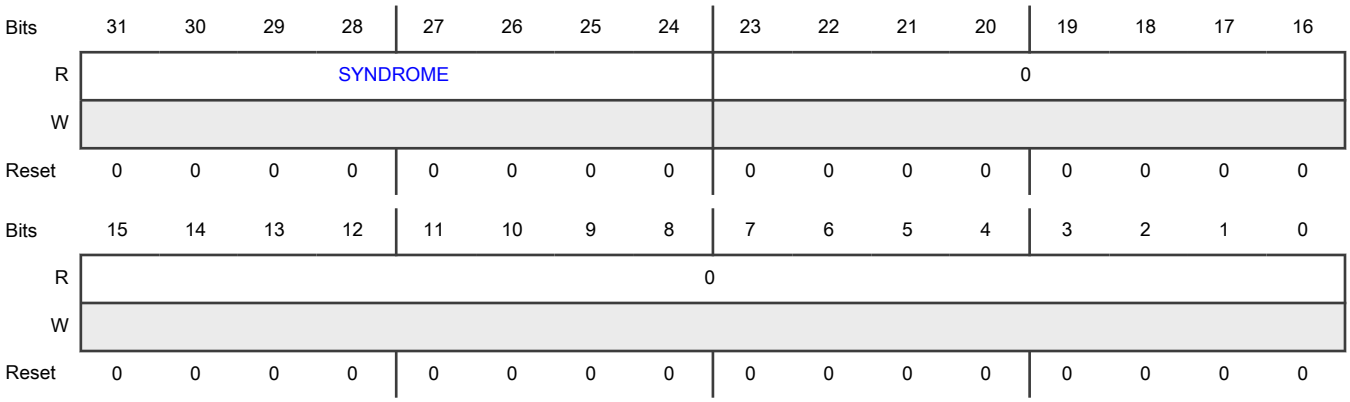
All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Function

The ERM Memory *n* Syndrome Register is a 32-bit register for capturing the calculated syndrome of the last ECC event on Memory*n*, where*n* denotes the memory channel. Any attempted write to SYN*n* is ignored. The syndrome value identifies the pertinent bit position on a correctable, single-bit data inversion or a non-correctable, single-bit address inversion. The syndrome value does not provide any additional diagnostic information on non-correctable, multi-bit inversions.

Diagram



Fields

Field	Function
31-24 SYNDROME	SYNDROME Memory*n* Syndrome — This field contains the ECC syndrome associated with the last recorded ECC event on Memory*n*.
23-0 —	Reserved

8.5.6 ERM Memory a Correctable Error Count Register (CORR_ERR_CNT0 - CORR_ERR_CNT1)

Offset

Register	Offset
CORR_ERR_CNT0	108h
CORR_ERR_CNT1	118h

Function

Each 32-bit ERM Memory *n* Correctable Error Count Register records the count value of the number of correctable ECC error events for Memory*n*, where*n* denotes the memory channel.

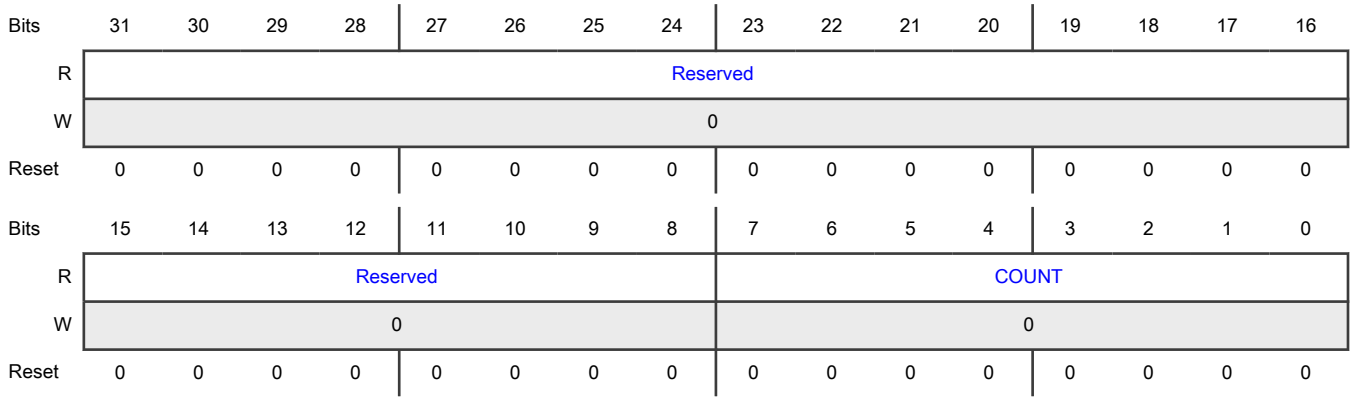
NOTE

Non-correctable errors are considered a serious fault, so the ERM does not provide any mechanism to count non-correctable errors. Only correctable errors are counted.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram



Fields

Field	Function
31-8 —	Reserved
7-0 COUNT	Memory n Correctable Error Count For each correctable error event, the ERM increments this field's error count value until the counter reaches its maximum value FFh. COUNT value will stop when it reaches maximum value FFh and will not wrap even if additional errors occur. Read this field to determine the correctable error count value so far. Write all zeros to this field to reset the counter. Writing a non-zero value has no effect.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Chapter 9
Signal Multiplexing

9.1 Introduction

Pins have several functions available via signal multiplexing to optimize functionality in small packages. This chapter illustrates which chip's signals are multiplexed on which external pin.

See MCX-C15_16_pinout to know how the pins available on this chip are configured. This spreadsheet also provides the available chip packages and pinout diagrams.

9.1.1 Signal multiplexing constraints

- A given module signal must be assigned to a maximum of one package pin. Do not program the same function to more than one pin.
- To ensure the best signal timing for a given peripheral's interface, choose the pins in closest proximity to each other.

9.2 Signal multiplexing and pin assignments

The signal multiplexing and pin assignments are provided in the following table and an Excel file is also available: The Port Control Module is responsible for selecting which ALT functionality is available on each pin.

Table 64. Signal multiplexing and pin assignments

Pin Name	H-PQFN48	LQFP48	H-PQFN32	H-PQFN24	H-PQFN16	Pinmux assignment	Pad settings	Alternate functions
P1_8	1	1	1	24	--	**ALT0** - P1_8 **ALT1** - FREQME_CLK_IN0 **ALT2** - LPUART1_RXD **ALT3** - LPI3C0_SDA **ALT4** - CT_INP8 **ALT5** - CT0_MAT2	**IO Supply** - VDD **Pad type** - HD **Default** - DIS	**ISP** - I2C_SDA **VDD SYS** - WUU0_IN10
P1_9	2	2	2	1	--	**ALT0** - P1_9 **ALT1** - FREQME_CLK_IN1 **ALT2** - LPUART1_TXD **ALT3** - LPI2C0_SCL **ALT4** - CT_INP9 **ALT5** - CT0_MAT3	**IO Supply** - VDD **Pad type** - HD **Default** - DIS	**ISP** - I2C_SCL
P1_10	3	--	--	--	--	**ALT0** - P1_10 **ALT2** - LPUART1_RTS_B **ALT3** - LPI2C0_SDAS **ALT4** - CT1_MAT0	**IO Supply** - VDD **Pad type** - SLOW **Default** - DIS	**ANALOG** - ADC0_A8
P1_11	4	--	--	--	--	**ALT0** - P1_11 **ALT1** - TRIG_OUT2 **ALT2** - LPUART1_CTS_B **ALT3** - LPI2C0_SCLS	**IO Supply** - VDD **Pad type** - SLOW **Default** - DIS	**ANALOG** - ADC0_A9 **VDD SYS** - WUU0_IN11

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table 64. Signal multiplexing and pin assignments (continued)

Pin Name	H-PQFN48	LQFP48	H-PQFN32	H-PQFN24	H-PQFN16	Pinmux assignment	Pad settings	Alternate functions
						ALT4 - CT1_MAT1		
P1_29	5	3	3	2	16	**ALT0** - P1_29 **ALT1** - RESET_B **ALT2** - SPC_LPREQ	**IO Supply** - VDD **Pad type** - RST **Default** - ALT1	**VDD SYS** - RESET_B
P1_30	6	4	4	3	1	**ALT0** - P1_30 **ALT1** - TRIG_OUT3 **ALT3** - LPI2C0_SDA **ALT4** - CT_INP16	**IO Supply** - VDD **Pad type** - SLOW **Default** - DIS	**ANALOG** - XTAL_32K
P1_31	7	5	5	4	2	**ALT0** - P1_31 **ALT1** - TRIG_IN4 **ALT2** - CLKIN **ALT3** - LPI2C0_SCL **ALT4** - CT_INP17	**IO Supply** - VDD **Pad type** - SLOW **Default** - DIS	**ANALOG** - EXTAL_32K
VSS	0	6	0	0	0	--	**IO Supply** - VDD	--
VDD_ANA	8	7	6	5	3	--	**IO Supply** - VDD	--
VDD	9	8	7	5	3	--	**IO Supply** - VDD	--
P2_0	10	9	--	--	--	**ALT0** - P2_0 **ALT1** - TRIG_IN6 **ALT2** - LPUART0_RXD **ALT4** - CT_INP16 **ALT5** - CT0_MAT0	**IO Supply** - VDD **Pad type** - SLOW **Default** - DIS	**ANALOG** - ADC0_A0 **VDD SYS** - WUU0_IN18
P2_1	11	10	--	--	--	**ALT0** - P2_1 **ALT1** - TRIG_IN7 **ALT2** - LPUART0_TXD **ALT4** - CT_INP17 **ALT5** - CT0_MAT1	**IO Supply** - VDD **Pad type** - SLOW **Default** - DIS	**ANALOG** - ADC0_A1
P2_2	12	11	8	6	4	**ALT0** - P2_2 **ALT1** - TRIG_IN6 **ALT2** - LPUART0_RTS_B **ALT3** - LPUART2_TXD **ALT4** - CT_INP12 **ALT5** - CT0_MAT2	**IO Supply** - VDD **Pad type** - SLOW **Default** - DIS	**ANALOG** - ADC0_A4/CMP0_IN0
P2_3	13	12	9	7	5	**ALT0** - P2_3 **ALT1** - TRIG_IN7 **ALT2** - LPUART0_CTS_B	**IO Supply** - VDD **Pad type** - SLOW **Default** - DIS	**ANALOG** - ADC0_A2/ OPAMP0_EXT_REF 1

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table 64. Signal multiplexing and pin assignments (continued)

Pin Name	H-PQFN48	LQFP48	H-PQFN32	H-PQFN24	H-PQFN16	Pinmux assignment	Pad settings	Alternate functions
						ALT3 - LPUART2_RXD **ALT4** - CT_INP13 **ALT5** - CT0_MAT3 **ALT9** - OPAMP0_REF_EXT_REF1_EN		**VDD SYS** - WUU0_IN19
P2_6	14	13	--	--	--	**ALT0** - P2_6 **ALT1** - TRIG_OUT4 **ALT2** - LPSPI0_PCS1 **ALT3** - LPUART3_RXD **ALT4** - CT_INP18 **ALT5** - CT1_MAT2	**IO Supply** - VDD **Pad type** - SLOW **Default** - DIS	--
P2_7	15	14	10	8	--	**ALT0** - P2_7 **ALT1** - TRIG_IN5 **ALT3** - LPUART3_TXD **ALT4** - CT_INP19 **ALT5** - CT1_MAT3 **ALT9** - OPAMP0_REF_EXT_REF0_EN	**IO Supply** - VDD **Pad type** - SLOW **Default** - DIS	**ANALOG** - VREFI/ ADC0_A7/ OPAMP0_EXT_REF 0
P2_12	16	15	11	9	6	**ALT0** - P2_12 **ALT2** - LPSPI0_SCK **ALT3** - LPUART1_RXD **ALT5** - CT0_MAT0	**IO Supply** - VDD **Pad type** - SLOW **Default** - DIS	**ANALOG** - ADC0_A5/ OPAMP0_INP0 **VDD SYS** - WUU0_IN20
P2_13	17	16	12	10	--	**ALT0** - P2_13 **ALT1** - TRIG_IN8 **ALT2** - LPSPI0_SDO **ALT3** - LPUART1_TXD **ALT5** - CT0_MAT1	**IO Supply** - VDD **Pad type** - SLOW **Default** - DIS	**ANALOG** - OPAMP0_INN
P2_16	18	17	13	11	--	**ALT0** - P2_16 **ALT2** - LPSPI0_SDI **ALT3** - LPUART1_RTS_B **ALT5** - CT0_MAT2 **ALT9** - OPAMP0_OUT_EN	**IO Supply** - VDD **Pad type** - SLOW **Default** - DIS	**ANALOG** - ADC0_A6/ OPAMP0_OUT/ CMP0_IN4P
P2_17	19	18	14	12	--	**ALT0** - P2_17 **ALT1** - TRIG_IN9 **ALT2** - LPSPI0_PCS0 **ALT3** - LPUART1_CTS_B	**IO Supply** - VDD **Pad type** - SLOW **Default** - DIS	**ANALOG** - OPAMP0_INP1

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table 64. Signal multiplexing and pin assignments (continued)

Pin Name	H-PQFN48	LQFP48	H-PQFN32	H-PQFN24	H-PQFN16	Pinmux assignment	Pad settings	Alternate functions
						ALT5 - CT0_MAT3		
P2_20	20	19	--	--	--	**ALT0** - P2_20 **ALT1** - TRIG_IN8 **ALT2** - LPSPI0_PCS2 **ALT4** - CT1_MAT0	**IO Supply** - VDD **Pad type** - SLOW **Default** - DIS	--
P2_21	21	20	--	--	--	**ALT0** - P2_21 **ALT1** - TRIG_IN9 **ALT2** - LPSPI0_PCS3 **ALT4** - CT1_MAT1	**IO Supply** - VDD **Pad type** - SLOW **Default** - DIS	--
P2_27	--	21	--	--	--	**ALT0** - P2_27 **ALT1** - TRIG_IN2 **ALT2** - LPUART3_RXD	**IO Supply** - VDD **Pad type** - SLOW **Default** - DIS	--
P2_28	22	22	--	--	--	**ALT0** - P2_28 **ALT1** - TRIG_OUT1 **ALT2** - LPUART3_TXD	**IO Supply** - VDD **Pad type** - SLOW **Default** - DIS	--
P3_31	23	23	--	--	--	**ALT0** - P3_31 **ALT1** - TRIG_IN10 **ALT4** - CT0_MAT3	**IO Supply** - VDD **Pad type** - SLOW **Default** - DIS	**ANALOG** - ADC0_A12 **VDD SYS** - LPTMR0_ALT2
P3_30	24	24	--	--	--	**ALT0** - P3_30 **ALT1** - TRIG_OUT6 **ALT4** - CT0_MAT2	**IO Supply** - VDD **Pad type** - SLOW **Default** - DIS	**ANALOG** - ADC0_A13
P3_29	25	25	15	13	7	**ALT0** - P3_29 **ALT1** - ISPMODE_N **ALT2** - CLKIN **ALT4** - CT_INP3	**IO Supply** - VDD **Pad type** - SLOW **Default** - ALT1	**ISP** - ISPMODE_N **ANALOG** - ADC0_A14 **VDD SYS** - WUU0_IN27
P3_28	26	26	16	--	--	**ALT0** - P3_28 **ALT1** - TRIG_IN11 **ALT2** - LPI2C0_SDA **ALT3** - LPUART3_RXD **ALT4** - CT_INP12	**IO Supply** - VDD **Pad type** - SLOW **Default** - DIS	**VDD SYS** - WUU0_IN26
P3_27	27	27	17	--	--	**ALT0** - P3_27 **ALT1** - TRIG_OUT7 **ALT2** - LPI2C0_SCL **ALT3** - LPUART3_TXD	**IO Supply** - VDD **Pad type** - SLOW **Default** - DIS	--

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table 64. Signal multiplexing and pin assignments (continued)

Pin Name	H-PQFN48	LQFP48	H-PQFN32	H-PQFN24	H-PQFN16	Pinmux assignment	Pad settings	Alternate functions
						ALT4 - CT_INP13 **ALT5** - PWM0_X2		
P3_14	28	--	--	--	--	**ALT0** - P3_14 **ALT4** - CT_INP6 **ALT5** - PWM0_X2	**IO Supply** - VDD **Pad type** - SLOW **Default** - DIS	**VDD SYS** - WUU0_IN25
P3_13	29	28	--	--	--	**ALT0** - P3_13 **ALT3** - LPUART2_RXD **ALT4** - CT1_MAT3 **ALT5** - PWM0_X1	**IO Supply** - VDD **Pad type** - SLOW **Default** - DIS	--
P3_12	30	29	--	--	--	**ALT0** - P3_12 **ALT3** - LPUART2_TXD **ALT4** - CT1_MAT2 **ALT5** - PWM0_X0	**IO Supply** - VDD **Pad type** - SLOW **Default** - DIS	--
P3_11	31	30	18	14	8	**ALT0** - P3_11 **ALT1** - TRIG_IN6 **ALT2** - LPSPI0_PCS0 **ALT3** - LPUART1_CTS_B **ALT4** - CT1_MAT1 **ALT5** - PWM0_B2	**IO Supply** - VDD **Pad type** - SLOW **Default** - DIS	**VDD SYS** - WUU0_IN24
P3_10	32	31	19	15	9	**ALT0** - P3_10 **ALT1** - TRIG_IN5 **ALT2** - LPSPI0_SCK **ALT3** - LPUART1_RTS_B **ALT4** - CT1_MAT0 **ALT5** - PWM0_A2	**IO Supply** - VDD **Pad type** - SLOW **Default** - DIS	--
P3_9	33	32	20	16	10	**ALT0** - P3_9 **ALT1** - TRIG_IN4 **ALT2** - LPSPI0_SDI **ALT3** - LPUART1_TXD **ALT4** - CT_INP5 **ALT5** - PWM0_B1	**IO Supply** - VDD **Pad type** - SLOW **Default** - DIS	--
P3_8	34	33	21	17	11	**ALT0** - P3_8 **ALT1** - TRIG_IN3 **ALT2** - LPSPI0_SDO **ALT3** - LPUART1_RXD **ALT4** - CT_INP4	**IO Supply** - VDD **Pad type** - SLOW **Default** - DIS	**VDD SYS** - WUU0_IN23

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table 64. Signal multiplexing and pin assignments (continued)

Pin Name	H-PQFN48	LQFP48	H-PQFN32	H-PQFN24	H-PQFN16	Pinmux assignment	Pad settings	Alternate functions
						ALT5 - PWM0_A1 **ALT12** - CLKOUT		
P3_1	35	34	22	18	--	**ALT0** - P3_1 **ALT1** - TRIG_IN1 **ALT3** - LPI2C0_SDA **ALT4** - CT_INP17 **ALT5** - PWM0_B0 **ALT12** - FREQME_CLK_OUT0	**IO Supply** - VDD **Pad type** - HD **Default** - DIS	--
P3_0	36	35	23	19	--	**ALT0** - P3_0 **ALT1** - TRIG_IN0 **ALT3** - LPI2C0_SCL **ALT4** - CT_INP16 **ALT5** - PWM0_A0	**IO Supply** - VDD **Pad type** - HD **Default** - DIS	**VDD SYS** - WUU0_IN22
VSS	0	36	0	0	0	--	**IO Supply** - VDD	--
VDD	37	37	24	--	--	--	**IO Supply** - VDD	--
P0_0	38	38	25	20	12	**ALT0** - P0_0 **ALT1** - TMS/SWDIO **ALT2** - LPUART0_RTS_B **ALT3** - LPSPI0_PCS0 **ALT4** - CT_INP0	**IO Supply** - VDD **Pad type** - SLOW **Default** - ALT1	--
P0_1	39	39	26	21	13	**ALT0** - P0_1 **ALT1** - TCLK/SWCLK **ALT2** - LPUART0_CTS_B **ALT3** - LPSPI0_SDI **ALT4** - CT_INP1	**IO Supply** - VDD **Pad type** - SLOW **Default** - ALT1	--
P0_2	40	40	27	22	14	**ALT0** - P0_2 **ALT1** - TDO **ALT2** - LPUART0_RXD **ALT3** - LPSPI0_SCK **ALT4** - CT0_MAT0	**IO Supply** - VDD **Pad type** - SLOW **Default** - ALT1	**ISP** - UART_RXD
P0_3	41	41	28	23	15	**ALT0** - P0_3 **ALT1** - TDI **ALT2** - LPUART0_TXD **ALT3** - LPSPI0_SDO **ALT4** - CT0_MAT1 **ALT8** - CMP0_OUT	**IO Supply** - VDD **Pad type** - SLOW **Default** - ALT1	**ISP** - UART_TXD

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table 64. Signal multiplexing and pin assignments (continued)

Pin Name	H-PQFN48	LQFP48	H-PQFN32	H-PQFN24	H-PQFN16	Pinmux assignment	Pad settings	Alternate functions
P0_6	42	42	--	--	--	**ALT0** - P0_6 **ALT2** - LPI2C0_HREQ **ALT3** - LPSPI0_PCS1 **ALT4** - CT_INP2 **ALT12** - CLKOUT	**IO Supply** - VDD **Pad type** - SLOW **Default** - DIS	**ANALOG** - ADC0_A15
P0_16	43	43	--	--	--	**ALT0** - P0_16 **ALT2** - LPI2C0_SDA **ALT3** - LPSPI0_PCS2 **ALT4** - CT0_MAT0	**IO Supply** - VDD **Pad type** - SLOW **Default** - DIS	**VDD SYS** - WUU0_IN2
P0_17	44	44	--	--	--	**ALT0** - P0_17 **ALT2** - LPI2C0_SCL **ALT3** - LPSPI0_PCS3 **ALT4** - CT0_MAT1	**IO Supply** - VDD **Pad type** - SLOW **Default** - DIS	--
P1_0	45	45	29	--	--	**ALT0** - P1_0 **ALT1** - TRIG_IN0 **ALT2** - LPSPI0_SDO **ALT3** - LPI2C0_SDA **ALT4** - CT_INP4 **ALT5** - CT0_MAT2	**IO Supply** - VDD **Pad type** - MED+I2C_FILT **Default** - DIS	**ANALOG** - ADC0_A16/ CMP0_IN3 **VDD SYS** - WUU0_IN6/ LPTMR0_ALT3
P1_1	46	46	30	--	--	**ALT0** - P1_1 **ALT1** - TRIG_IN1 **ALT2** - LPSPI0_SCK **ALT3** - LPI2C0_SCL **ALT4** - CT_INP5 **ALT5** - CT0_MAT3	**IO Supply** - VDD **Pad type** - MED+I2C_FILT **Default** - DIS	**ANALOG** - ADC0_A17
P1_2	47	47	31	--	--	**ALT0** - P1_2 **ALT1** - TRIG_OUT0 **ALT2** - LPSPI0_SDI **ALT3** - LPI2C0_SDAS **ALT4** - CT1_MAT0 **ALT5** - CT_INP0	**IO Supply** - VDD **Pad type** - MED **Default** - DIS	**ANALOG** - ADC0_A18
P1_3	48	48	32	--	--	**ALT0** - P1_3 **ALT1** - TRIG_OUT1 **ALT2** - LPSPI0_PCS0 **ALT3** - LPI2C0_SCLS **ALT4** - CT1_MAT1	**IO Supply** - VDD **Pad type** - SLOW **Default** - DIS	**ANALOG** - ADC0_A19/ CMP0_IN1 **VDD SYS** - WUU0_IN7

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table 64. Signal multiplexing and pin assignments (continued)

Pin Name	H-PQFN48	LQFP48	H-PQFN32	H-PQFN24	H-PQFN16	Pinmux assignment	Pad settings	Alternate functions
						ALT5 - CT_INP1		

NOTE

1. +I2C_FILT in Pad Type represents that I2C filter is implemented on the pin. PFE bit is implemented in Pin Control register of the pin.

2. HD in Pad Type represents that the pin can support up to 20 mA drive strength. I2C filter is implemented on the pin. PFE bit is implemented in Pin Control register of the pin.

3. DIS in default column represents that the pin's input buffer is disabled by default

4. RST pads support passive filter and 1 MΩ pull resistor. PFE and PV bits are implemented in Pin Control register of the pin.

5. PE, PS, SRE, ODE and DSE are supported in Pin Control register of all types of IO.

6. HD pads support two DSE bits in Pin Control register of the pin.

7. SLOW in Pad Type represents the IO supports 25 MHz. MED in Pad Type represents the IO supports 50 MHz.

8. In order to connect pin signals of OPAMP0_OUT, OPAMP0_EXT_REF_0 and OPAMP0_EXT_REF_1 to OPAMP, you need to configure the related pins to ALT9.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Chapter 10
Port Control (PORT)

10.1 Chip-specific PORT information

Table 65. Reference links to related information

Topic	Related module	Reference
Full description	PORT	PORT
System memory map		Memory map
Clocking		Clock distribution
Power Management		Power Management
Signal multiplexing	Port control	Signal multiplexing

10.1.1 Module instances

This chip has four instances of the port module, PORT0, PORT1, PORT2 and PORT3.

10.1.2 Normal I/O pin structure

The following diagram shows the structure of normal I/O pin.

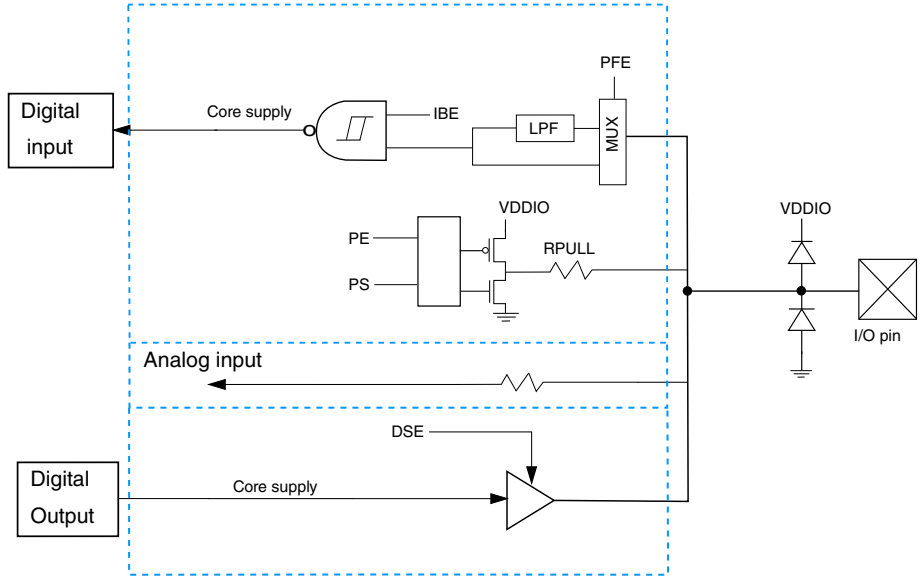


Figure 12. Basic structure of an I/O port

10.1.3 Calibration

In this chip, only the pins with the pad type of MED (see Table 64) support calibration feature which is controlled by PORT0.CALIB0 and PORT0.CALIB1.

© 2026 NXP B.V. All rights reserved.

10.2 Overview

PORT provides support for pad control functions. You can configure most functions independently for each pin, in the 32-bit port, and affect the pin regardless of its pin multiplexing state.

There exists a single instance of the PORT module for each port, and not all pins within each port are implemented on a specific chip.

10.2.1 Block diagram

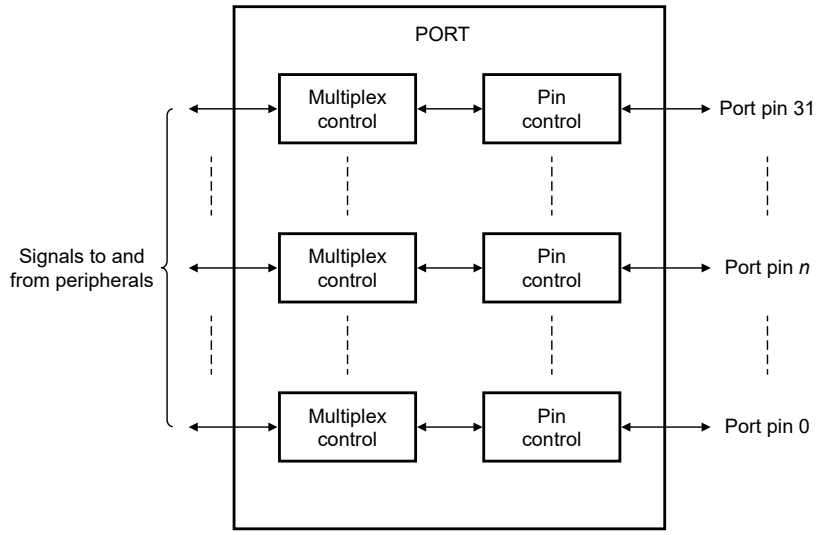


Figure 13. Block diagram

10.2.2 Features

- Individual pull control fields with pullup, pulldown, and pull-disable support
- Individual drive strength fields supporting high and low drive strengths
- Individual slew rate fields supporting fast and slow slew rates
- Individual PCR*n*[ODE] fields that enable and disable individual open drain outputs
- Digital input inversion to optionally invert the digital input
- Digital PCR*n*[IBE] fields to enables digital input buffer.
- Individual PCR*n*[MUX] fields supporting GPIO and multiple chip-specific analog and digital functions

10.3 Functional description

10.3.1 Pin control

Each port pin has a corresponding Pin Control Register (PCR) associated with it that helps you configure the following functions for each pin within the 32-bit port:

- Pullup or pulldown enable
- Drive strength configuration
- Slew rate configuration
- Open drain enable

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

- Digital input inversion
- Software configuration lock
- Pin multiplexing mode

These functions apply across all digital pin multiplexing modes, and individual peripherals do not override the configuration in PCR*n* unless otherwise noted. For example, if an I^2C function is enabled on a pin, it does not override the pullup or open drain configuration for that pin.

PCR*n*[LK] allows the configuration for each pin to be locked until the next system reset. When locked, writes to the lower half of that PCR*n* are ignored, although a bus error is not generated on an attempted write to a locked register.

The configuration of each PCR*n* is retained when the PORT module is disabled.

When you configure a pin in a digital pin multiplexing mode, the input buffer for that pin is enabled, allowing the pin state to be read via the corresponding GPIO.PDIR or allowing a pin interrupt or DMA request to be generated. If a pin is always floating when its input buffer is enabled, it can cause an increase in power consumption. This situation must be avoided. A pin can be floating because of an input pin that is not connected or an output pin that is tristated (output buffer is disabled).

Enabling the internal pull resistor (or implementing an external pull resistor) ensures that a pin does not float when its input buffer is enabled. The internal pull resistor is automatically disabled whenever the output buffer is enabled, allowing PCR*n*[PE] to remain 1. Configuring Pin Multiplexing mode to disabled or analog (PCR*n*[MUX] = 0) disables the pin's input buffer and results in lowest power consumption.

10.3.2 Global pin control

Global Pin Control Low (GPCLR) and Global Pin Control High (GPCHR) allow a single register write to update the lower 16 bits of PCR*n* for up to 16 pins, all with the same value. You cannot write to locked registers by using Global Pin Control Low (GPCLR) and Global Pin Control High (GPCHR).

Global Pin Control Low (GPCLR) and Global Pin Control High (GPCHR) enable you to quickly configure multiple pins within the same port and with the same peripheral function. These are write-only registers that always read as 0.

10.3.3 Clocking

This module has no clocking considerations.

10.3.4 Calibration

The combination of Calibration 0 (CALIB0) and Calibration 1 (CALIB1) controls the drive strength of a pin for the port and PCR*n*[DSE] for the pin.

Calibration 0 (CALIB0) and Calibration 1 (CALIB1) represent two driver configurations. If PCR*n*[DSE] = 0, the configuration in Calibration 0 (CALIB0) is used. Likewise, if PCR*n*[DSE] = 1, the configuration in Calibration 1 (CALIB1) is used.

The pulldown and pullup drivers have eight segments. The value of NCAL[5:3] represents the number of pulldown segments that are enabled when driving low, and the value of PCAL[5:3] represents the number of pullup segments that turn on when driving high. Writing 000b to these fields causes one segment to turn on, while writing 111b causes all eight segments to turn on. Upon reset, Calibration 0 (CALIB0) becomes 1 to create approximately 50 Ω driver at 3.3 V and 25 °C (3 is a typical value) and Calibration 1 (CALIB1) becomes 1 to create approximately 50 Ω driver at 1.8 V and 25 °C (7 is the typical value).

The three LSBs of CALIB0[NCAL], CALIB1[NCAL], CALIB0[PCAL], and CALIB1[PCAL] do not affect the driver's strength. Instead, they are included to improve precision when calibrating to a different impedance value. This can be done by using the following formula:

Calibration = (G × (CAL0 + 4) – 1024) >> 8, where G is a binary code representing the desired conductance and CAL0 is one of the 6-bit fields in Calibration 0 (CALIB0) and Calibration 1 (CALIB1). The value of G scales linearly with the intended conductance; G = 0 corresponds to 0 S (an open circuit), and G = 256 corresponds to 0.02 S (50 Ω). For example, the following operation configures the driver to be approximately 0.03 S (33 Ω) at 3.3 V when PCR*n*[DSE] = 1:

The value of CALIB1[NCAL] = (384 × (the value of CALIB0[NCAL] + 4) – 1024) >> 8;

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

The value of CALIB1[PCAL] = (384 × (the value of CALIB0[PCAL] + 4) – 1024) >> 8;

Do not overflow or underflow NCAL[5:3] or PCAL[5:3]; otherwise, this operation may have an unintended result. You can configure the drive strength for 50 Ω at 2.5 V and 25 °C by averaging the initial NCAL codes in Calibration 0 (CALIB0) and Calibration 1 (CALIB1), and averaging the initial PCAL codes in Calibration 0 (CALIB0) and Calibration 1 (CALIB1). However, the 2.5 V configuration is not as accurate or precise as the original 1.8 V and 3.3 V configuration because the drive strength is not perfectly linear with supply voltage.

NOTE

This calibration feature may not apply to all PORT pins. See the chip-specific PORT information for pins that support calibration.

10.4 Initialization

To initialize PORT, perform the following procedure:

1. Initialize the pin functions:
   - Initialize single pin functions by writing appropriate values to PCR*n*.
   - Initialize multiple pins (up to 16) with the same configuration by writing appropriate values to Global Pin Control Low (GPCLR) or Global Pin Control High (GPCHR).
2. Lock the configuration for a given pin, by writing 1 to PCR*n*[LK], so that it cannot be changed until the next reset.

10.5 Application information

10.6 Memory map and register definition

Any read or write access to the PORT memory space, outside the valid memory map, results in a bus error. All register accesses complete with zero wait states.

10.6.1 PORT register descriptions

10.6.1.1 PORT memory map

PORT0 base address: 400B_C000h

PORT1 base address: 400B_D000h

PORT2 base address: 400B_E000h

PORT3 base address: 400B_F000h

Offset	Register	Width (In bits)	Access	Reset value
0h	Version ID (VERID)	32	R	0200_0000h
10h	Global Pin Control Low (GPCLR)	32	RW	0000_0000h
14h	Global Pin Control High (GPCHR)	32	RW	0000_0000h
20h	Configuration (CONFIG)	32	RW	0000_0000h
60h	Calibration 0 (CALIB0)	32	RW	See section
64h	Calibration 1 (CALIB1)	32	RW	See section

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Offset	Register	Width (In bits)	Access	Reset value
80h	Pin Control 0 (PCR0)	32	RW	See section
84h	Pin Control 1 (PCR1)	32	RW	See section
88h	Pin Control 2 (PCR2)	32	RW	See section
8Ch	Pin Control 3 (PCR3)	32	RW	See section
98h	Pin Control 6 (PCR6)	32	RW	0000_0000h
9Ch	Pin Control 7 (PCR7)	32	RW	0000_0000h
A0h - A4h	Pin Control a (PCR8 - PCR9)	32	RW	0000_0000h
A8h - ACh	Pin Control a (PCR10 - PCR11)	32	RW	0000_0000h
B0h	Pin Control 12 (PCR12)	32	RW	0000_0000h
B4h	Pin Control 13 (PCR13)	32	RW	0000_0000h
B8h	Pin Control 14 (PCR14)	32	RW	0000_0000h
C0h	Pin Control 16 (PCR16)	32	RW	0000_0000h
C4h	Pin Control 17 (PCR17)	32	RW	0000_0000h
D0h	Pin Control 20 (PCR20)	32	RW	0000_0000h
D4h	Pin Control 21 (PCR21)	32	RW	0000_0000h
ECh	Pin Control 27 (PCR27)	32	RW	0000_0000h
F0h	Pin Control 28 (PCR28)	32	RW	0000_0000h
F4h	Pin Control 29 (PCR29)	32	RW	See section
F8h - FCh	Pin Control a (PCR30 - PCR31)	32	RW	0000_0000h

10.6.1.2 Version ID (VERID)

Offset

Register	Offset
VERID	0h

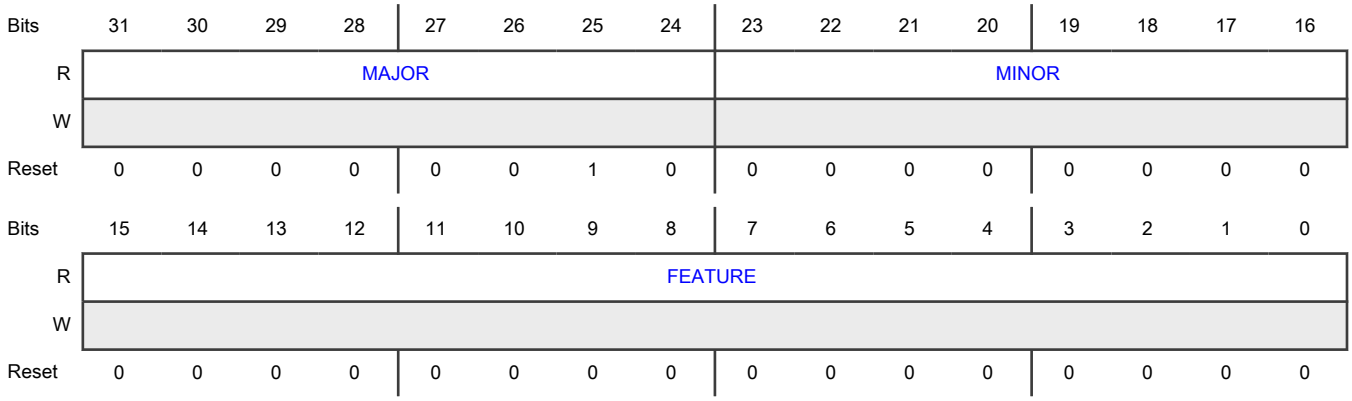
Function

Specifies the version number and feature number of the chip.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram



Fields

Field	Function
31-24 MAJOR	Major Version Number Indicates the major version number for the specification.
23-16 MINOR	Minor Version Number Indicates the minor version number for the specification.
15-0 FEATURE	Feature Specification Number Indicates the feature set number. 0000_0000_0000_0000b - Basic implementation

10.6.1.3 Global Pin Control Low (GPCLR)

Offset

Register	Offset
GPCLR	10h

Function

Controls writes to the PCR15–PCR0 registers.

NOTE

This register supports only 32-bit writes.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
W	GPWE 15	GPWE 14	GPWE 13	GPWE 12	GPWE 11	GPWE 10	GPWE 9	GPWE 8	GPWE 7	GPWE 6	GPWE 5	GPWE 4	GPWE 3	GPWE 2	GPWE 1	GPWE 0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															
W	GPWD															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-16 GPWEn	Global Pin Write Enable Configures the corresponding lower 16-bit field of PCR*n* to be updated with the value in the GPWD field. If a selected PCR is locked, the write to that register is ignored. 0b - Not updated 1b - Updated
15-0 GPWD	Global Pin Write Data Is written to PCR*n*[15:0] if GPWE*n* = 1.

10.6.1.4 Global Pin Control High (GPCHR)

Offset

Register	Offset
GPCHR	14h

Function

Controls writes to the PCR31–PCR16 registers.

NOTE

This register supports only 32-bit writes.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
W	GPWE 31	GPWE 30	GPWE 29	GPWE 28	GPWE 27	GPWE 26	GPWE 25	GPWE 24	GPWE 23	GPWE 22	GPWE 21	GPWE 20	GPWE 19	GPWE 18	GPWE 17	GPWE 16
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															
W	GPWD															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-16 GPWEn	Global Pin Write Enable Configures the corresponding lower 16-bit field of PCR*n* to be updated with the value in the GPWD field. If a selected PCR is locked, write to that register is ignored. 0b - Not updated 1b - Updated
15-0 GPWD	Global Pin Write Data Is written to PCR*n*[15:0] if GPWE*n* = 1.

10.6.1.5 Configuration (CONFIG)

Offset

Register	Offset
CONFIG	20h

Function

Configures the port voltage range.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W	0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved															RANGE
W	0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-1 —	Reserved
0 RANGE	Port Voltage Range Configures the port voltage range. 0b - 1.71 V–3.6 V 1b - 2.70 V–3.6 V

10.6.1.6 Calibration 0 (CALIB0)

Offset

Register	Offset
CALIB0	60h

Function

Stores calibration values for the PMOS and NMOS output drivers when PCR*n*[DSE] = 0.

NOTE

Each module instance supports a different number of registers.

Instance	Register supported	Register not supported
PORT0	CALIB0	—
PORT1	—	CALIB0
PORT2	—	CALIB0

Table continues on the next page...

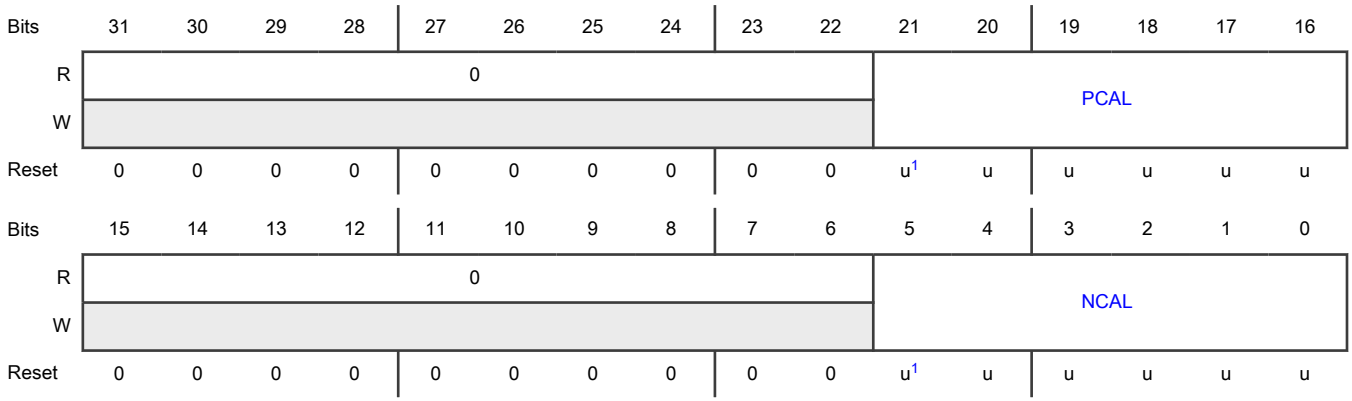
All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Instance	Register supported	Register not supported
PORT3	—	CALIB0

Diagram



1. Reset values are loaded out of IFR.

Fields

Field	Function
31-22 —	Reserved
21-16 PCAL	Calibration of PMOS Output Driver
15-6 —	Reserved
5-0 NCAL	Calibration of NMOS Output Driver

10.6.1.7 Calibration 1 (CALIB1)

Offset

Register	Offset
CALIB1	64h

Function

Stores calibration values for the PMOS and NMOS output drivers when PCR*n*[DSE] = 1.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

NOTE

Each module instance supports a different number of registers.

Instance	Register supported	Register not supported
PORT0	CALIB1	—
PORT1	—	CALIB1
PORT2	—	CALIB1
PORT3	—	CALIB1

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0										PCAL					
W																
Reset	0	0	0	0	0	0	0	0	0	0	u[1]	u	u	u	u	u

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0										NCAL					
W																
Reset	0	0	0	0	0	0	0	0	0	0	u[1]	u	u	u	u	u

1. Reset values are loaded out of IFR.

Fields

Field	Function
31-22 —	Reserved
21-16 PCAL	Calibration of PMOS Output Driver
15-6 —	Reserved
5-0 NCAL	Calibration of NMOS Output Driver

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

10.6.1.8 Pin Control 0 (PCR0)

Offset

Register	Offset
PCR0	80h

Function

Configures pin control features on each pin.

NOTE

Do not modify pin configuration registers associated with pins that are unavailable in your selected package. All unbonded pins unavailable in your package default to the Disabled state for lowest power consumption.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	See Register reset values.															

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	LK	0	INV	IBE	MUX				DSE1	DSE	ODE	PFE	SRE	0	PE	PS
W																
Reset	See Register reset values.															

Register reset values

Register	Reset value
PCR0	PORT0: 0000_1143h PORT1–PORT3: 0000_0000h

Fields

Field	Function
31-16 —	Reserved
15 LK	Lock Register Locks this PCR. When a PCR*n* is locked, its fields cannot be updated until the next reset.

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	0b - Does not lock 1b - Locks
14 —	Reserved
13 INV	Invert Input Inverts the digital input. 0b - Does not invert 1b - Inverts
12 IBE	Input Buffer Enable Enables digital input buffer. **NOTE** When a pin is configured as digital peripheral (including GPIO) through PCR[MUX]: • If the peripheral is input or bidirectional function, this bit must be 1. • If the peripheral is output only, the value of this bit doesn't matter. However, if it's 1, you can get the pin state through reading PDIR register. 0b - Disables 1b - Enables
11-8 MUX	Pin Multiplex Control Configures the multiplexing slots on each pin. Not all pins support all pin multiplexing slots. Unimplemented pin multiplexing slots are reserved. Unimplemented pin multiplexing slots can result in different behaviors, if the unimplemented slot is phantomed (because the module is phantomed on that die) versus unimplemented on the die. The corresponding pin is configured according to the following pin multiplexing slots: **NOTE** This field is not supported in every instance. The following table includes only supported registers.

Instance	Field supported in	Field not supported in
PORT0	PCR0[10–8]	PCR0[11]
PORT1	PCR0	—
PORT2	PCR0[10–8]	PCR0[11]

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	Instance / Field supported in / Field not supported in
	PORT3 / PCR0[10–8] / PCR0[11]
	NOTE The descriptions of the field settings vary by module instance.
	Instance / Field value and description
	PORT0 PORT2 PORT3 000b - Alternative 0 (GPIO) 001b - Alternative 1 (chip-specific) 010b - Alternative 2 (chip-specific) 011b - Alternative 3 (chip-specific) 100b - Alternative 4 (chip-specific) 101b - Alternative 5 (chip-specific) 110b - Alternative 6 (chip-specific) 111b - Alternative 7 (chip-specific)
	PORT1 0000b - Alternative 0 (GPIO) 0001b - Alternative 1 (chip-specific) 0010b - Alternative 2 (chip-specific) 0011b - Alternative 3 (chip-specific) 0100b - Alternative 4 (chip-specific) 0101b - Alternative 5 (chip-specific) 0110b - Alternative 6 (chip-specific) 0111b - Alternative 7 (chip-specific) 1000b - Alternative 8 (chip-specific) 1001b - Alternative 9 (chip-specific) 1010b - Alternative 10 (chip-specific) 1011b - Alternative 11 (chip-specific) 1100b - Alternative 12 (chip-specific) 1101b - Alternative 13 (chip-specific)
7 DSE1	Drive Strength Enable Configures the drive strength on each pin. The drive strength configuration is valid for all digital pin multiplexing modes.

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

<table>
<tr><th>Field</th><th>Function</th></tr>
<tr><td></td><td>

If this field = 0, normal drive strength is configured on the corresponding pin.
If this field = 1, double drive strength is configured on the corresponding pin.

NOTE
This field is not supported in every instance. The following table includes only supported registers.
<table>
<tr><th>Instance</th><th>Field supported in</th><th>Field not supported in</th></tr>
<tr><td>PORT0</td><td>—</td><td>PCR0</td></tr>
<tr><td>PORT1</td><td>—</td><td>PCR0</td></tr>
<tr><td>PORT2</td><td>—</td><td>PCR0</td></tr>
<tr><td>PORT3</td><td>PCR0</td><td>—</td></tr>
</table>
0b - Normal
1b - Double
</td></tr>
<tr><td>6
DSE</td><td>Drive Strength Enable
Configures drive strength, low or high, on each pin.
The drive strength configuration is valid for all digital pin multiplexing modes.

When this field = 0, low drive strength is configured on the corresponding pin, if the pin is configured as a digital output.
When this field = 1, high drive strength is configured on the corresponding pin, if the pin is configured as a digital output.

0b - Low
1b - High</td></tr>
<tr><td>5
ODE</td><td>Open Drain Enable
Enables open drain output on each pin.
The open drain configuration is valid for all digital pin multiplexing modes.

When this field = 0, the open drain output is disabled on the corresponding pin.
When this field = 1, the open drain output is enabled on the corresponding pin, if the pin is configured as a digital output.

0b - Disables
1b - Enables</td></tr>
<tr><td>4
PFE</td><td>Passive Filter Enable
Enables passive input filter on each pin.</td></tr>
</table>

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

<table>
<tr><th>Field</th><th>Function</th></tr>
<tr><td></td><td>The passive filter configuration is valid for all digital pin multiplexing modes.
• When this field = 0, the passive input filter is disabled on the corresponding pin.
• When this field = 1, the passive input filter is enabled on the corresponding pin, if the pin is configured as a digital input.
See the chip's data sheet for filter characteristics.
NOTE
This field is not supported in every instance. The following table includes only supported registers.
<table><tr><th>Instance</th><th>Field supported in</th><th>Field not supported in</th></tr><tr><td>PORT0</td><td>—</td><td>PCR0</td></tr><tr><td>PORT1</td><td>PCR0</td><td>—</td></tr><tr><td>PORT2</td><td>—</td><td>PCR0</td></tr><tr><td>PORT3</td><td>PCR0</td><td>—</td></tr></table>
0b - Disables
1b - Enables</td></tr>
<tr><td>3
SRE</td><td>Slew Rate Enable
Configures the slew rate feature, fast or slow, on each corresponding pin.
The slew rate configuration is valid for all digital pin multiplexing modes.
0b - Fast
1b - Slow</td></tr>
<tr><td>2
—</td><td>Reserved</td></tr>
<tr><td>1
PE</td><td>Pull Enable
Enables the internal pull resistor.
This configuration is valid for all digital pin multiplexing modes.
• When this field = 0, the internal pull resistor is not enabled on the corresponding pin.
• When this field = 1, the internal pull resistor is enabled on the corresponding pin, if the pin is configured as a digital input.
0b - Disables
1b - Enables</td></tr>
</table>

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
0 PS	Pull Select Enables the internal pullup or pulldown resistor. This configuration is valid for all digital pin multiplexing modes. • When this field = 0, the internal pulldown resistor is enabled on the corresponding pin, if the corresponding PCR*n*.PE field = 1. • When this field = 1, the internal pullup resistor is enabled on the corresponding pin, if the corresponding PCR*n*.PE field = 1. 0b - Enables internal pulldown resistor 1b - Enables internal pullup resistor

10.6.1.9 Pin Control 1 (PCR1)

Offset

Register	Offset
PCR1	84h

Function

Configures pin control features on each pin.

NOTE

Do not modify pin configuration registers associated with pins that are unavailable in your selected package. All unbonded pins unavailable in your package default to the Disabled state for lowest power consumption.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	See Register reset values.															

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	LK	0	INV	IBE	MUX				DSE1	DSE	ODE	PFE	SRE	0	PE	PS
W																
Reset	See Register reset values.															

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Register reset values

Register	Reset value
PCR1	PORT0: 0000_1102h PORT1–PORT3: 0000_0000h

Fields

Field	Function
31-16 —	Reserved
15 LK	Lock Register Locks this PCR. When a PCR*n* is locked, its fields cannot be updated until the next reset. 0b - Does not lock 1b - Locks
14 —	Reserved
13 INV	Invert Input Inverts the digital input. 0b - Does not invert 1b - Inverts
12 IBE	Input Buffer Enable Enables digital input buffer. **NOTE** When a pin is configured as digital peripheral (including GPIO) through PCR[MUX]: • If the peripheral is input or bidirectional function, this bit must be 1. • If the peripheral is output only, the value of this bit doesn't matter. However, if it's 1, you can get the pin state through reading PDIR register. 0b - Disables 1b - Enables
11-8 MUX	Pin Multiplex Control Configures the multiplexing slots on each pin. Not all pins support all pin multiplexing slots. Unimplemented pin multiplexing slots are reserved. Unimplemented pin multiplexing slots can result in different behaviors, if the unimplemented slot is phantomed (because the module is phantomed on that die) versus unimplemented on the die.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	The corresponding pin is configured according to the following pin multiplexing slots: **NOTE** This field is not supported in every instance. The following table includes only supported registers.

Instance	Field supported in	Field not supported in
PORT0	PCR1[10–8]	PCR1[11]
PORT1	PCR1	—
PORT2	PCR1[10–8]	PCR1[11]
PORT3	PCR1	—

NOTE
The descriptions of the field settings vary by module instance.

Instance	Field value and description
PORT0 PORT2	000b - Alternative 0 (GPIO) 001b - Alternative 1 (chip-specific) 010b - Alternative 2 (chip-specific) 011b - Alternative 3 (chip-specific) 100b - Alternative 4 (chip-specific) 101b - Alternative 5 (chip-specific) 110b - Alternative 6 (chip-specific) 111b - Alternative 7 (chip-specific)
PORT1 PORT3	0000b - Alternative 0 (GPIO) 0001b - Alternative 1 (chip-specific) 0010b - Alternative 2 (chip-specific) 0011b - Alternative 3 (chip-specific) 0100b - Alternative 4 (chip-specific) 0101b - Alternative 5 (chip-specific) 0110b - Alternative 6 (chip-specific) 0111b - Alternative 7 (chip-specific) 1000b - Alternative 8 (chip-specific)

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	<table><tr><th>Instance</th><th>Field value and description</th></tr><tr><td></td><td>1001b - Alternative 9 (chip-specific) 1010b - Alternative 10 (chip-specific) 1011b - Alternative 11 (chip-specific) 1100b - Alternative 12 (chip-specific) 1101b - Alternative 13 (chip-specific)</td></tr></table>
7 DSE1	Drive Strength Enable Configures the drive strength on each pin. The drive strength configuration is valid for all digital pin multiplexing modes. • If this field = 0, normal drive strength is configured on the corresponding pin. • If this field = 1, double drive strength is configured on the corresponding pin. **NOTE** This field is not supported in every instance. The following table includes only supported registers. <table><tr><th>Instance</th><th>Field supported in</th><th>Field not supported in</th></tr><tr><td>PORT0</td><td>—</td><td>PCR1</td></tr><tr><td>PORT1</td><td>—</td><td>PCR1</td></tr><tr><td>PORT2</td><td>—</td><td>PCR1</td></tr><tr><td>PORT3</td><td>PCR1</td><td>—</td></tr></table> 0b - Normal 1b - Double
6 DSE	Drive Strength Enable Configures drive strength, low or high, on each pin. The drive strength configuration is valid for all digital pin multiplexing modes. • When this field = 0, low drive strength is configured on the corresponding pin, if the pin is configured as a digital output. • When this field = 1, high drive strength is configured on the corresponding pin, if the pin is configured as a digital output. 0b - Low

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	1b - High
5 ODE	Open Drain Enable Enables open drain output on each pin. The open drain configuration is valid for all digital pin multiplexing modes. • When this field = 0, the open drain output is disabled on the corresponding pin. • When this field = 1, the open drain output is enabled on the corresponding pin, if the pin is configured as a digital output. 0b - Disables 1b - Enables
4 PFE	Passive Filter Enable Enables passive input filter on each pin. The passive filter configuration is valid for all digital pin multiplexing modes. • When this field = 0, the passive input filter is disabled on the corresponding pin. • When this field = 1, the passive input filter is enabled on the corresponding pin, if the pin is configured as a digital input. See the chip's data sheet for filter characteristics. **NOTE** This field is not supported in every instance. The following table includes only supported registers. Instance / Field supported in / Field not supported in: PORT0 / — / PCR1 PORT1 / PCR1 / — PORT2 / — / PCR1 PORT3 / PCR1 / — 0b - Disables 1b - Enables
3 SRE	Slew Rate Enable Configures the slew rate feature, fast or slow, on each corresponding pin. The slew rate configuration is valid for all digital pin multiplexing modes. 0b - Fast

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	1b - Slow
2 —	Reserved
1 PE	Pull Enable Enables the internal pull resistor. This configuration is valid for all digital pin multiplexing modes. • When this field = 0, the internal pull resistor is not enabled on the corresponding pin. • When this field = 1, the internal pull resistor is enabled on the corresponding pin, if the pin is configured as a digital input. 0b - Disables 1b - Enables
0 PS	Pull Select Enables the internal pullup or pulldown resistor. This configuration is valid for all digital pin multiplexing modes. • When this field = 0, the internal pulldown resistor is enabled on the corresponding pin, if the corresponding PCR*n*.PE field = 1. • When this field = 1, the internal pullup resistor is enabled on the corresponding pin, if the corresponding PCR*n*.PE field = 1. 0b - Enables internal pulldown resistor 1b - Enables internal pullup resistor

10.6.1.10 Pin Control 2 (PCR2)

Offset

Register	Offset
PCR2	88h

Function

Configures pin control features on each pin.

NOTE

Do not modify pin configuration registers associated with pins that are unavailable in your selected package. All unbonded pins unavailable in your package default to the Disabled state for lowest power consumption.

NOTE

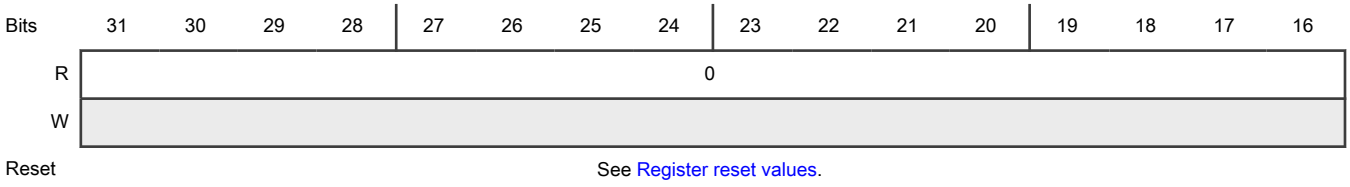
Each module instance supports a different number of registers.

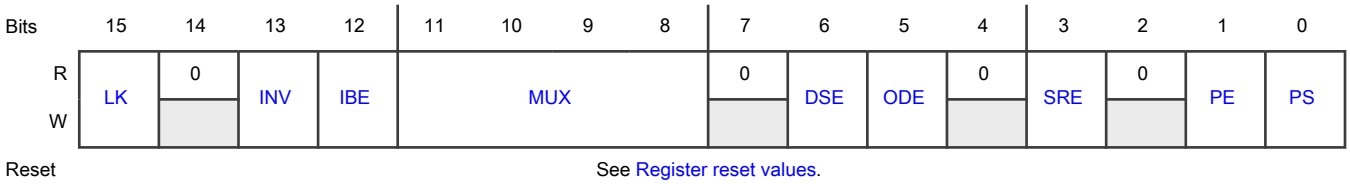
All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Instance	Register supported	Register not supported
PORT0	PCR2	—
PORT1	PCR2	—
PORT2	PCR2	—
PORT3	—	PCR2

Diagram





Register reset values

Register	Reset value
PCR2	PORT0: 0000_0140h PORT1,PORT2: 0000_0000h PORT3: Register not supported

Fields

Field	Function
31-16 —	Reserved
15 LK	Lock Register Locks this PCR. When a PCR*n* is locked, its fields cannot be updated until the next reset. 0b - Does not lock 1b - Locks
14	Reserved

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
—	
13 INV	Invert Input Inverts the digital input. 0b - Does not invert 1b - Inverts
12 IBE	Input Buffer Enable Enables digital input buffer. **NOTE** When a pin is configured as digital peripheral (including GPIO) through PCR[MUX]: • If the peripheral is input or bidirectional function, this bit must be 1. • If the peripheral is output only, the value of this bit doesn't matter. However, if it's 1, you can get the pin state through reading PDIR register. 0b - Disables 1b - Enables
11-8 MUX	Pin Multiplex Control Configures the multiplexing slots on each pin. Not all pins support all pin multiplexing slots. Unimplemented pin multiplexing slots are reserved. Unimplemented pin multiplexing slots can result in different behaviors, if the unimplemented slot is phantomed (because the module is phantomed on that die) versus unimplemented on the die. The corresponding pin is configured according to the following pin multiplexing slots: **NOTE** This field is not supported in every instance. The following table includes only supported registers.

Instance	Field supported in	Field not supported in
PORT0	PCR2	—
PORT1	PCR2[10–8]	PCR2[11]
PORT2	PCR2[10–8]	PCR2[11]

NOTE

The descriptions of the field settings vary by module instance.

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	<table><tr><th>Instance</th><th>Field value and description</th></tr><tr><td>PORT0</td><td>0000b - Alternative 0 (GPIO) 0001b - Alternative 1 (chip-specific) 0010b - Alternative 2 (chip-specific) 0011b - Alternative 3 (chip-specific) 0100b - Alternative 4 (chip-specific) 0101b - Alternative 5 (chip-specific) 0110b - Alternative 6 (chip-specific) 0111b - Alternative 7 (chip-specific) 1000b - Alternative 8 (chip-specific) 1001b - Alternative 9 (chip-specific) 1010b - Alternative 10 (chip-specific) 1011b - Alternative 11 (chip-specific) 1100b - Alternative 12 (chip-specific) 1101b - Alternative 13 (chip-specific)</td></tr><tr><td>PORT1 PORT2</td><td>000b - Alternative 0 (GPIO) 001b - Alternative 1 (chip-specific) 010b - Alternative 2 (chip-specific) 011b - Alternative 3 (chip-specific) 100b - Alternative 4 (chip-specific) 101b - Alternative 5 (chip-specific) 110b - Alternative 6 (chip-specific) 111b - Alternative 7 (chip-specific)</td></tr></table>
7 —	Reserved
6 DSE	Drive Strength Enable Configures drive strength, low or high, on each pin. The drive strength configuration is valid for all digital pin multiplexing modes. • When this field = 0, low drive strength is configured on the corresponding pin, if the pin is configured as a digital output. • When this field = 1, high drive strength is configured on the corresponding pin, if the pin is configured as a digital output.

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	0b - Low 1b - High
5 ODE	Open Drain Enable Enables open drain output on each pin. The open drain configuration is valid for all digital pin multiplexing modes. • When this field = 0, the open drain output is disabled on the corresponding pin. • When this field = 1, the open drain output is enabled on the corresponding pin, if the pin is configured as a digital output. 0b - Disables 1b - Enables
4 —	Reserved
3 SRE	Slew Rate Enable Configures the slew rate feature, fast or slow, on each corresponding pin. The slew rate configuration is valid for all digital pin multiplexing modes. 0b - Fast 1b - Slow
2 —	Reserved
1 PE	Pull Enable Enables the internal pull resistor. This configuration is valid for all digital pin multiplexing modes. • When this field = 0, the internal pull resistor is not enabled on the corresponding pin. • When this field = 1, the internal pull resistor is enabled on the corresponding pin, if the pin is configured as a digital input. 0b - Disables 1b - Enables
0 PS	Pull Select Enables the internal pullup or pulldown resistor. This configuration is valid for all digital pin multiplexing modes. • When this field = 0, the internal pulldown resistor is enabled on the corresponding pin, if the corresponding PCR*n*.PE field = 1.

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	• When this field = 1, the internal pullup resistor is enabled on the corresponding pin, if the corresponding PCR*n*.PE field = 1. 0b - Enables internal pulldown resistor 1b - Enables internal pullup resistor

10.6.1.11 Pin Control 3 (PCR3)

Offset

Register	Offset
PCR3	8Ch

Function

Configures pin control features on each pin.

NOTE

Do not modify pin configuration registers associated with pins that are unavailable in your selected package. All unbonded pins unavailable in your package default to the Disabled state for lowest power consumption.

NOTE

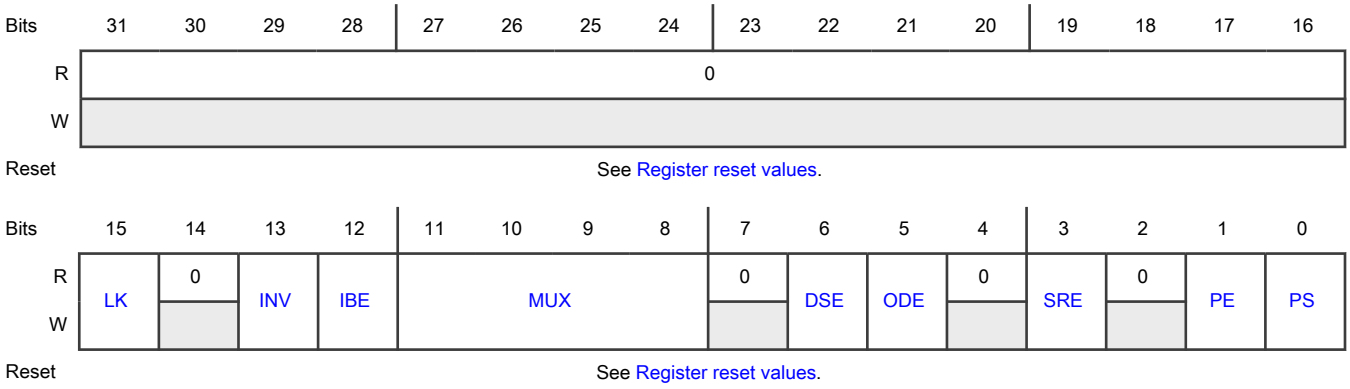
Each module instance supports a different number of registers.

Instance	Register supported	Register not supported
PORT0	PCR3	—
PORT1	PCR3	—
PORT2	PCR3	—
PORT3	—	PCR3

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram



Register reset values

Register	Reset value
PCR3	PORT0: 0000_1103h PORT1,PORT2: 0000_0000h PORT3: Register not supported

Fields

Field	Function
31-16 —	Reserved
15 LK	Lock Register Locks this PCR. When a PCR*n* is locked, its fields cannot be updated until the next reset. 0b - Does not lock 1b - Locks
14 —	Reserved
13 INV	Invert Input Inverts the digital input. 0b - Does not invert 1b - Inverts
12 IBE	Input Buffer Enable Enables digital input buffer.

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

<table>
<tr><th>Field</th><th>Function</th></tr>
<tr><td></td><td>
NOTE
When a pin is configured as digital peripheral (including GPIO) through PCR[MUX]:
• If the peripheral is input or bidirectional function, this bit must be 1.
• If the peripheral is output only, the value of this bit doesn’t matter. However, if it's 1, you can get the pin state through reading PDIR register.
0b - Disables
1b - Enables
</td></tr>
<tr><td>11-8
MUX</td><td>
Pin Multiplex Control
Configures the multiplexing slots on each pin.
Not all pins support all pin multiplexing slots. Unimplemented pin multiplexing slots are reserved.
Unimplemented pin multiplexing slots can result in different behaviors, if the unimplemented slot is phantomed (because the module is phantomed on that die) versus unimplemented on the die.
The corresponding pin is configured according to the following pin multiplexing slots:
NOTE
This field is not supported in every instance. The following table includes only supported registers.
<table>
<tr><th>Instance</th><th>Field supported in</th><th>Field not supported in</th></tr>
<tr><td>PORT0</td><td>PCR3</td><td>—</td></tr>
<tr><td>PORT1</td><td>PCR3[10–8]</td><td>PCR3[11]</td></tr>
<tr><td>PORT2</td><td>PCR3</td><td>—</td></tr>
</table>
NOTE
The descriptions of the field settings vary by module instance.
<table>
<tr><th>Instance</th><th>Field value and description</th></tr>
<tr><td>PORT0
PORT2</td><td>
0000b - Alternative 0 (GPIO)
0001b - Alternative 1 (chip-specific)
0010b - Alternative 2 (chip-specific)
0011b - Alternative 3 (chip-specific)
0100b - Alternative 4 (chip-specific)
0101b - Alternative 5 (chip-specific)
0110b - Alternative 6 (chip-specific)
</td></tr>
</table>
</td></tr>
</table>

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

<table>
<tr><th>Field</th><th>Function</th></tr>
<tr><td></td><td>
<table>
<tr><th>Instance</th><th>Field value and description</th></tr>
<tr><td></td><td>0111b - Alternative 7 (chip-specific)
1000b - Alternative 8 (chip-specific)
1001b - Alternative 9 (chip-specific)
1010b - Alternative 10 (chip-specific)
1011b - Alternative 11 (chip-specific)
1100b - Alternative 12 (chip-specific)
1101b - Alternative 13 (chip-specific)</td></tr>
<tr><td>PORT1</td><td>000b - Alternative 0 (GPIO)
001b - Alternative 1 (chip-specific)
010b - Alternative 2 (chip-specific)
011b - Alternative 3 (chip-specific)
100b - Alternative 4 (chip-specific)
101b - Alternative 5 (chip-specific)
110b - Alternative 6 (chip-specific)
111b - Alternative 7 (chip-specific)</td></tr>
</table>
</td></tr>
<tr><td>7
—</td><td>Reserved</td></tr>
<tr><td>6
DSE</td><td>Drive Strength Enable
Configures drive strength, low or high, on each pin.
The drive strength configuration is valid for all digital pin multiplexing modes.
• When this field = 0, low drive strength is configured on the corresponding pin, if the pin is configured as a digital output.
• When this field = 1, high drive strength is configured on the corresponding pin, if the pin is configured as a digital output.
0b - Low
1b - High</td></tr>
<tr><td>5
ODE</td><td>Open Drain Enable
Enables open drain output on each pin.
The open drain configuration is valid for all digital pin multiplexing modes.
• When this field = 0, the open drain output is disabled on the corresponding pin.</td></tr>
</table>

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	• When this field = 1, the open drain output is enabled on the corresponding pin, if the pin is configured as a digital output. 0b - Disables 1b - Enables
4 —	Reserved
3 SRE	Slew Rate Enable Configures the slew rate feature, fast or slow, on each corresponding pin. The slew rate configuration is valid for all digital pin multiplexing modes. 0b - Fast 1b - Slow
2 —	Reserved
1 PE	Pull Enable Enables the internal pull resistor. This configuration is valid for all digital pin multiplexing modes. • When this field = 0, the internal pull resistor is not enabled on the corresponding pin. • When this field = 1, the internal pull resistor is enabled on the corresponding pin, if the pin is configured as a digital input. 0b - Disables 1b - Enables
0 PS	Pull Select Enables the internal pullup or pulldown resistor. This configuration is valid for all digital pin multiplexing modes. • When this field = 0, the internal pulldown resistor is enabled on the corresponding pin, if the corresponding PCR*n*.PE field = 1. • When this field = 1, the internal pullup resistor is enabled on the corresponding pin, if the corresponding PCR*n*.PE field = 1. 0b - Enables internal pulldown resistor 1b - Enables internal pullup resistor

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

10.6.1.12 Pin Control 6 (PCR6)

Offset

Register	Offset
PCR6	98h

Function

Configures pin control features on each pin.

NOTE

Do not modify pin configuration registers associated with pins that are unavailable in your selected package. All unbonded pins unavailable in your package default to the Disabled state for lowest power consumption.

NOTE

Each module instance supports a different number of registers.

Instance	Register supported	Register not supported
PORT0	PCR6	—
PORT1	—	PCR6
PORT2	PCR6	—
PORT3	—	PCR6

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	LK	0	INV	IBE	MUX				0	DSE	ODE	0	SRE	0	PE	PS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-16 —	Reserved

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
15 LK	Lock Register Locks this PCR. When a PCR*n* is locked, its fields cannot be updated until the next reset. 0b - Does not lock 1b - Locks
14 —	Reserved
13 INV	Invert Input Inverts the digital input. 0b - Does not invert 1b - Inverts
12 IBE	Input Buffer Enable Enables digital input buffer. **NOTE** When a pin is configured as digital peripheral (including GPIO) through PCR[MUX]: • If the peripheral is input or bidirectional function, this bit must be 1. • If the peripheral is output only, the value of this bit doesn't matter. However, if it's 1, you can get the pin state through reading PDIR register. 0b - Disables 1b - Enables
11-8 MUX	Pin Multiplex Control Configures the multiplexing slots on each pin. Not all pins support all pin multiplexing slots. Unimplemented pin multiplexing slots are reserved. Unimplemented pin multiplexing slots can result in different behaviors, if the unimplemented slot is phantomed (because the module is phantomed on that die) versus unimplemented on the die. The corresponding pin is configured according to the following pin multiplexing slots: **NOTE** This field is not supported in every instance. The following table includes only supported registers.

Instance	Field supported in	Field not supported in
PORT0	PCR6	—

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	<table><tr><th>Instance</th><th>Field supported in</th><th>Field not supported in</th></tr><tr><td>PORT2</td><td>PCR6[10–8]</td><td>PCR6[11]</td></tr></table> **NOTE** The descriptions of the field settings vary by module instance. <table><tr><th>Instance</th><th>Field value and description</th></tr><tr><td>PORT0</td><td>0000b - Alternative 0 (GPIO) 0001b - Alternative 1 (chip-specific) 0010b - Alternative 2 (chip-specific) 0011b - Alternative 3 (chip-specific) 0100b - Alternative 4 (chip-specific) 0101b - Alternative 5 (chip-specific) 0110b - Alternative 6 (chip-specific) 0111b - Alternative 7 (chip-specific) 1000b - Alternative 8 (chip-specific) 1001b - Alternative 9 (chip-specific) 1010b - Alternative 10 (chip-specific) 1011b - Alternative 11 (chip-specific) 1100b - Alternative 12 (chip-specific) 1101b - Alternative 13 (chip-specific)</td></tr><tr><td>PORT2</td><td>000b - Alternative 0 (GPIO) 001b - Alternative 1 (chip-specific) 010b - Alternative 2 (chip-specific) 011b - Alternative 3 (chip-specific) 100b - Alternative 4 (chip-specific) 101b - Alternative 5 (chip-specific) 110b - Alternative 6 (chip-specific) 111b - Alternative 7 (chip-specific)</td></tr></table>
7 —	Reserved

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
6 DSE	Drive Strength Enable Configures drive strength, low or high, on each pin. The drive strength configuration is valid for all digital pin multiplexing modes. • When this field = 0, low drive strength is configured on the corresponding pin, if the pin is configured as a digital output. • When this field = 1, high drive strength is configured on the corresponding pin, if the pin is configured as a digital output. 0b - Low 1b - High
5 ODE	Open Drain Enable Enables open drain output on each pin. The open drain configuration is valid for all digital pin multiplexing modes. • When this field = 0, the open drain output is disabled on the corresponding pin. • When this field = 1, the open drain output is enabled on the corresponding pin, if the pin is configured as a digital output. 0b - Disables 1b - Enables
4 —	Reserved
3 SRE	Slew Rate Enable Configures the slew rate feature, fast or slow, on each corresponding pin. The slew rate configuration is valid for all digital pin multiplexing modes. 0b - Fast 1b - Slow
2 —	Reserved
1 PE	Pull Enable Enables the internal pull resistor. This configuration is valid for all digital pin multiplexing modes. • When this field = 0, the internal pull resistor is not enabled on the corresponding pin. • When this field = 1, the internal pull resistor is enabled on the corresponding pin, if the pin is configured as a digital input. 0b - Disables

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	1b - Enables
0 PS	Pull Select Enables the internal pullup or pulldown resistor. This configuration is valid for all digital pin multiplexing modes. • When this field = 0, the internal pulldown resistor is enabled on the corresponding pin, if the corresponding PCR*n*.PE field = 1. • When this field = 1, the internal pullup resistor is enabled on the corresponding pin, if the corresponding PCR*n*.PE field = 1. 0b - Enables internal pulldown resistor 1b - Enables internal pullup resistor

10.6.1.13 Pin Control 7 (PCR7)

Offset

Register	Offset
PCR7	9Ch

Function

Configures pin control features on each pin.

NOTE

Do not modify pin configuration registers associated with pins that are unavailable in your selected package. All unbonded pins unavailable in your package default to the Disabled state for lowest power consumption.

NOTE

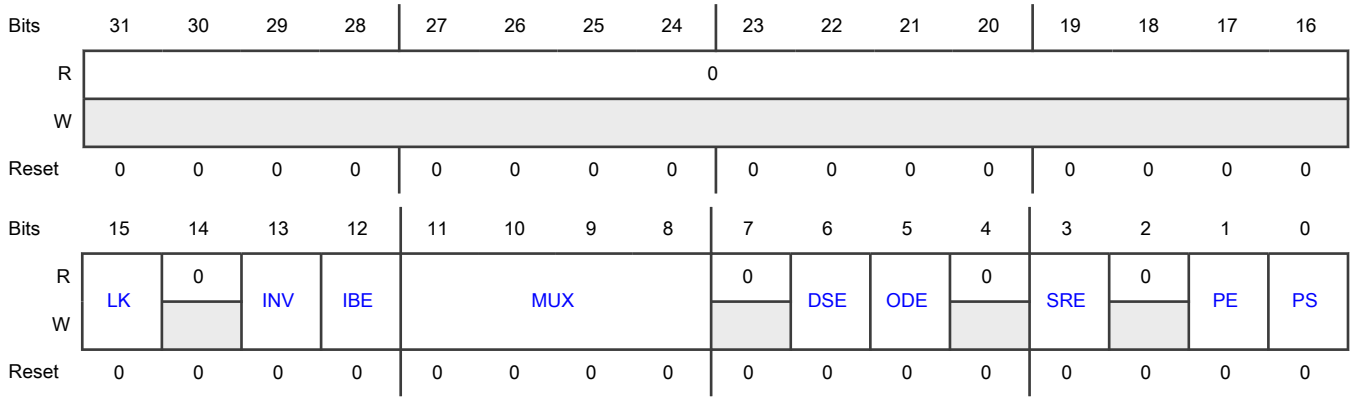
Each module instance supports a different number of registers.

Instance	Register supported	Register not supported
PORT0	—	PCR7
PORT1	—	PCR7
PORT2	PCR7	—
PORT3	—	PCR7

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram



Fields

Field	Function
31-16 —	Reserved
15 LK	Lock Register Locks this PCR. When a PCR*n* is locked, its fields cannot be updated until the next reset. 0b - Does not lock 1b - Locks
14 —	Reserved
13 INV	Invert Input Inverts the digital input. 0b - Does not invert 1b - Inverts
12 IBE	Input Buffer Enable Enables digital input buffer. **NOTE** When a pin is configured as digital peripheral (including GPIO) through PCR[MUX]: • If the peripheral is input or bidirectional function, this bit must be 1. • If the peripheral is output only, the value of this bit doesn't matter. However, if it's 1, you can get the pin state through reading PDIR register. 0b - Disables 1b - Enables

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
11-8 MUX	Pin Multiplex Control Configures the multiplexing slots on each pin. Not all pins support all pin multiplexing slots. Unimplemented pin multiplexing slots are reserved. Unimplemented pin multiplexing slots can result in different behaviors, if the unimplemented slot is phantomed (because the module is phantomed on that die) versus unimplemented on the die. The corresponding pin is configured according to the following pin multiplexing slots: 0000b - Alternative 0 (GPIO) 0001b - Alternative 1 (chip-specific) 0010b - Alternative 2 (chip-specific) 0011b - Alternative 3 (chip-specific) 0100b - Alternative 4 (chip-specific) 0101b - Alternative 5 (chip-specific) 0110b - Alternative 6 (chip-specific) 0111b - Alternative 7 (chip-specific) 1000b - Alternative 8 (chip-specific) 1001b - Alternative 9 (chip-specific) 1010b - Alternative 10 (chip-specific) 1011b - Alternative 11 (chip-specific) 1100b - Alternative 12 (chip-specific) 1101b - Alternative 13 (chip-specific)
7 —	Reserved
6 DSE	Drive Strength Enable Configures drive strength, low or high, on each pin. The drive strength configuration is valid for all digital pin multiplexing modes. • When this field = 0, low drive strength is configured on the corresponding pin, if the pin is configured as a digital output. • When this field = 1, high drive strength is configured on the corresponding pin, if the pin is configured as a digital output. 0b - Low 1b - High
5 ODE	Open Drain Enable Enables open drain output on each pin. The open drain configuration is valid for all digital pin multiplexing modes.

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	• When this field = 0, the open drain output is disabled on the corresponding pin. • When this field = 1, the open drain output is enabled on the corresponding pin, if the pin is configured as a digital output. 0b - Disables 1b - Enables
4 —	Reserved
3 SRE	Slew Rate Enable Configures the slew rate feature, fast or slow, on each corresponding pin. The slew rate configuration is valid for all digital pin multiplexing modes. 0b - Fast 1b - Slow
2 —	Reserved
1 PE	Pull Enable Enables the internal pull resistor. This configuration is valid for all digital pin multiplexing modes. • When this field = 0, the internal pull resistor is not enabled on the corresponding pin. • When this field = 1, the internal pull resistor is enabled on the corresponding pin, if the pin is configured as a digital input. 0b - Disables 1b - Enables
0 PS	Pull Select Enables the internal pullup or pulldown resistor. This configuration is valid for all digital pin multiplexing modes. • When this field = 0, the internal pulldown resistor is enabled on the corresponding pin, if the corresponding PCR*n*.PE field = 1. • When this field = 1, the internal pullup resistor is enabled on the corresponding pin, if the corresponding PCR*n*.PE field = 1. 0b - Enables internal pulldown resistor 1b - Enables internal pullup resistor

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

10.6.1.14 Pin Control a (PCR8 - PCR9)

Offset

Register	Offset
PCR8	A0h
PCR9	A4h

Function

Configures pin control features on each pin.

NOTE

Do not modify pin configuration registers associated with pins that are unavailable in your selected package. All unbonded pins unavailable in your package default to the Disabled state for lowest power consumption.

NOTE

Each module instance supports a different number of registers.

Instance	Register supported	Register not supported
PORT0	—	PCR8–PCR9
PORT1	PCR8–PCR9	—
PORT2	—	PCR8–PCR9
PORT3	PCR8–PCR9	—

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	LK	0	INV	IBE	MUX				DSE1	DSE	ODE	PFE	SRE	0	PE	PS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-16	Reserved

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
—	
15 LK	Lock Register Locks this PCR. When a PCR*n* is locked, its fields cannot be updated until the next reset. 0b - Does not lock 1b - Locks
14 —	Reserved
13 INV	Invert Input Inverts the digital input. 0b - Does not invert 1b - Inverts
12 IBE	Input Buffer Enable Enables digital input buffer. **NOTE** When a pin is configured as digital peripheral (including GPIO) through PCR[MUX]: • If the peripheral is input or bidirectional function, this bit must be 1. • If the peripheral is output only, the value of this bit doesn't matter. However, if it's 1, you can get the pin state through reading PDIR register. 0b - Disables 1b - Enables
11-8 MUX	Pin Multiplex Control Configures the multiplexing slots on each pin. Not all pins support all pin multiplexing slots. Unimplemented pin multiplexing slots are reserved. Unimplemented pin multiplexing slots can result in different behaviors, if the unimplemented slot is phantomed (because the module is phantomed on that die) versus unimplemented on the die. The corresponding pin is configured according to the following pin multiplexing slots: 0000b - Alternative 0 (GPIO) 0001b - Alternative 1 (chip-specific) 0010b - Alternative 2 (chip-specific) 0011b - Alternative 3 (chip-specific) 0100b - Alternative 4 (chip-specific)

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

<table>
<tr><th>Field</th><th>Function</th></tr>
<tr><td></td><td>0101b - Alternative 5 (chip-specific)
0110b - Alternative 6 (chip-specific)
0111b - Alternative 7 (chip-specific)
1000b - Alternative 8 (chip-specific)
1001b - Alternative 9 (chip-specific)
1010b - Alternative 10 (chip-specific)
1011b - Alternative 11 (chip-specific)
1100b - Alternative 12 (chip-specific)
1101b - Alternative 13 (chip-specific)</td></tr>
<tr><td>7
DSE1</td><td>Drive Strength Enable
Configures the drive strength on each pin.
The drive strength configuration is valid for all digital pin multiplexing modes.
• If this field = 0, normal drive strength is configured on the corresponding pin.
• If this field = 1, double drive strength is configured on the corresponding pin.
NOTE
This field is not supported in every instance. The following table includes only supported registers.
<table><tr><th>Instance</th><th>Field supported in</th><th>Field not supported in</th></tr><tr><td>PORT1</td><td>PCR8–PCR9</td><td>—</td></tr><tr><td>PORT3</td><td>—</td><td>PCR8–PCR9</td></tr></table>
0b - Normal
1b - Double</td></tr>
<tr><td>6
DSE</td><td>Drive Strength Enable
Configures drive strength, low or high, on each pin.
The drive strength configuration is valid for all digital pin multiplexing modes.
• When this field = 0, low drive strength is configured on the corresponding pin, if the pin is configured as a digital output.
• When this field = 1, high drive strength is configured on the corresponding pin, if the pin is configured as a digital output.
0b - Low
1b - High</td></tr>
</table>

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

<table>
<tr><th>Field</th><th>Function</th></tr>
<tr><td>5
ODE</td><td>Open Drain Enable
Enables open drain output on each pin.
The open drain configuration is valid for all digital pin multiplexing modes.
• When this field = 0, the open drain output is disabled on the corresponding pin.
• When this field = 1, the open drain output is enabled on the corresponding pin, if the pin is configured as a digital output.
0b - Disables
1b - Enables</td></tr>
<tr><td>4
PFE</td><td>Passive Filter Enable
Enables passive input filter on each pin.
The passive filter configuration is valid for all digital pin multiplexing modes.
• When this field = 0, the passive input filter is disabled on the corresponding pin.
• When this field = 1, the passive input filter is enabled on the corresponding pin, if the pin is configured as a digital input.
See the chip's data sheet for filter characteristics.
NOTE
This field is not supported in every instance. The following table includes only supported registers.
<table><tr><th>Instance</th><th>Field supported in</th><th>Field not supported in</th></tr><tr><td>PORT1</td><td>PCR8–PCR9</td><td>—</td></tr><tr><td>PORT3</td><td>—</td><td>PCR8–PCR9</td></tr></table>
0b - Disables
1b - Enables</td></tr>
<tr><td>3
SRE</td><td>Slew Rate Enable
Configures the slew rate feature, fast or slow, on each corresponding pin.
The slew rate configuration is valid for all digital pin multiplexing modes.
0b - Fast
1b - Slow</td></tr>
<tr><td>2
—</td><td>Reserved</td></tr>
<tr><td>1</td><td>Pull Enable</td></tr>
</table>

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
PE	Enables the internal pull resistor. This configuration is valid for all digital pin multiplexing modes. • When this field = 0, the internal pull resistor is not enabled on the corresponding pin. • When this field = 1, the internal pull resistor is enabled on the corresponding pin, if the pin is configured as a digital input. 0b - Disables 1b - Enables
0 PS	Pull Select Enables the internal pullup or pulldown resistor. This configuration is valid for all digital pin multiplexing modes. • When this field = 0, the internal pulldown resistor is enabled on the corresponding pin, if the corresponding PCR*n*.PE field = 1. • When this field = 1, the internal pullup resistor is enabled on the corresponding pin, if the corresponding PCR*n*.PE field = 1. 0b - Enables internal pulldown resistor 1b - Enables internal pullup resistor

10.6.1.15 Pin Control a (PCR10 - PCR11)

Offset

Register	Offset
PCR10	A8h
PCR11	ACh

Function

Configures pin control features on each pin.

NOTE

Do not modify pin configuration registers associated with pins that are unavailable in your selected package. All unbonded pins unavailable in your package default to the Disabled state for lowest power consumption.

NOTE

Each module instance supports a different number of registers.

Instance	Register supported	Register not supported
PORT0	—	PCR10–PCR11

Table continues on the next page...

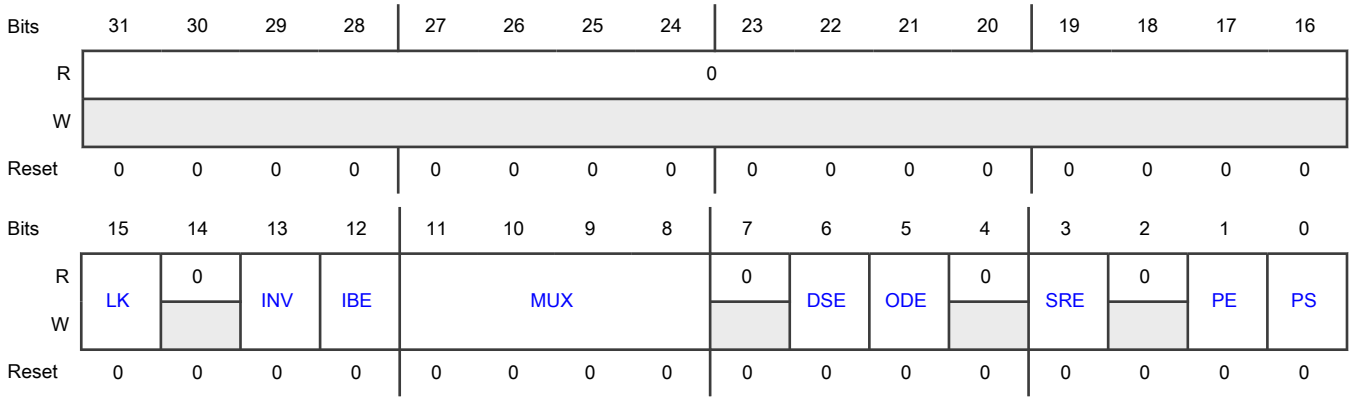
All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Instance	Register supported	Register not supported
PORT1	PCR10–PCR11	—
PORT2	—	PCR10–PCR11
PORT3	PCR10–PCR11	—

Diagram



Fields

Field	Function
31-16 —	Reserved
15 LK	Lock Register Locks this PCR. When a PCR*n* is locked, its fields cannot be updated until the next reset. 0b - Does not lock 1b - Locks
14 —	Reserved
13 INV	Invert Input Inverts the digital input. 0b - Does not invert 1b - Inverts
12	Input Buffer Enable

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
IBE	Enables digital input buffer. **NOTE** When a pin is configured as digital peripheral (including GPIO) through PCR[MUX]: • If the peripheral is input or bidirectional function, this bit must be 1. • If the peripheral is output only, the value of this bit doesn't matter. However, if it's 1, you can get the pin state through reading PDIR register. 0b - Disables 1b - Enables
11-8 MUX	Pin Multiplex Control Configures the multiplexing slots on each pin. Not all pins support all pin multiplexing slots. Unimplemented pin multiplexing slots are reserved. Unimplemented pin multiplexing slots can result in different behaviors, if the unimplemented slot is phantomed (because the module is phantomed on that die) versus unimplemented on the die. The corresponding pin is configured according to the following pin multiplexing slots: **NOTE** This field is not supported in every instance. The following table includes only supported registers.

Instance	Field supported in	Field not supported in
PORT1	PCR10–PCR11[10–8]	PCR10–PCR11[11]
PORT3	PCR10–PCR11	—

NOTE
The descriptions of the field settings vary by module instance.

Instance	Field value and description
PORT1	000b - Alternative 0 (GPIO) 001b - Alternative 1 (chip-specific) 010b - Alternative 2 (chip-specific) 011b - Alternative 3 (chip-specific) 100b - Alternative 4 (chip-specific) 101b - Alternative 5 (chip-specific) 110b - Alternative 6 (chip-specific)

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

<table>
<tr><th>Field</th><th>Function</th></tr>
<tr><td></td><td>
<table>
<tr><th>Instance</th><th>Field value and description</th></tr>
<tr><td></td><td>111b - Alternative 7 (chip-specific)</td></tr>
<tr><td>PORT3</td><td>0000b - Alternative 0 (GPIO)
0001b - Alternative 1 (chip-specific)
0010b - Alternative 2 (chip-specific)
0011b - Alternative 3 (chip-specific)
0100b - Alternative 4 (chip-specific)
0101b - Alternative 5 (chip-specific)
0110b - Alternative 6 (chip-specific)
0111b - Alternative 7 (chip-specific)
1000b - Alternative 8 (chip-specific)
1001b - Alternative 9 (chip-specific)
1010b - Alternative 10 (chip-specific)
1011b - Alternative 11 (chip-specific)
1100b - Alternative 12 (chip-specific)
1101b - Alternative 13 (chip-specific)</td></tr>
</table>
</td></tr>
<tr><td>7
—</td><td>Reserved</td></tr>
<tr><td>6
DSE</td><td>Drive Strength Enable
Configures drive strength, low or high, on each pin.
The drive strength configuration is valid for all digital pin multiplexing modes.
• When this field = 0, low drive strength is configured on the corresponding pin, if the pin is configured as a digital output.
• When this field = 1, high drive strength is configured on the corresponding pin, if the pin is configured as a digital output.
0b - Low
1b - High</td></tr>
<tr><td>5
ODE</td><td>Open Drain Enable
Enables open drain output on each pin.
The open drain configuration is valid for all digital pin multiplexing modes.
• When this field = 0, the open drain output is disabled on the corresponding pin.</td></tr>
</table>

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	• When this field = 1, the open drain output is enabled on the corresponding pin, if the pin is configured as a digital output. 0b - Disables 1b - Enables
4 —	Reserved
3 SRE	Slew Rate Enable Configures the slew rate feature, fast or slow, on each corresponding pin. The slew rate configuration is valid for all digital pin multiplexing modes. 0b - Fast 1b - Slow
2 —	Reserved
1 PE	Pull Enable Enables the internal pull resistor. This configuration is valid for all digital pin multiplexing modes. • When this field = 0, the internal pull resistor is not enabled on the corresponding pin. • When this field = 1, the internal pull resistor is enabled on the corresponding pin, if the pin is configured as a digital input. 0b - Disables 1b - Enables
0 PS	Pull Select Enables the internal pullup or pulldown resistor. This configuration is valid for all digital pin multiplexing modes. • When this field = 0, the internal pulldown resistor is enabled on the corresponding pin, if the corresponding PCR*n*.PE field = 1. • When this field = 1, the internal pullup resistor is enabled on the corresponding pin, if the corresponding PCR*n*.PE field = 1. 0b - Enables internal pulldown resistor 1b - Enables internal pullup resistor

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

10.6.1.16 Pin Control a (PCR12 - PCR16)

Offset

Register	Offset
PCR12	B0h
PCR13	B4h
PCR14	B8h
PCR16	C0h

Function

Configures pin control features on each pin.

NOTE

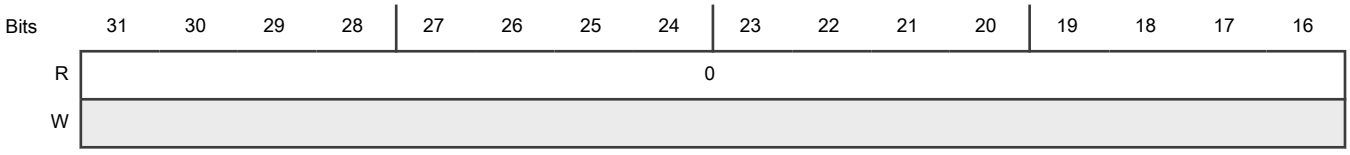
Do not modify pin configuration registers associated with pins that are unavailable in your selected package. All unbonded pins unavailable in your package default to the Disabled state for lowest power consumption.

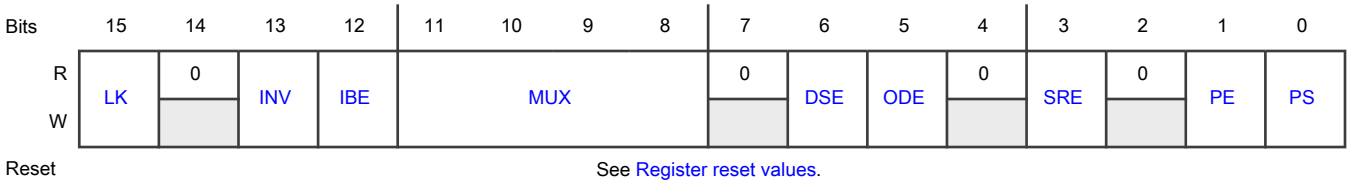
NOTE

Each module instance supports a different number of registers.

Instance	Register supported	Register not supported
PORT0	PCR16	PCR12–PCR14
PORT1	—	PCR12–PCR14 PCR16
PORT2	PCR12–PCR13 PCR16	PCR14
PORT3	PCR12–PCR14	PCR16

Diagram





All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Register reset values

Register	Reset value
PCR12–PCR13	PORT0,PORT1: Register not supported PORT2,PORT3: 0000_0000h
PCR14	0000_0000h
PCR15	—
PCR16	PORT0: 0000_0000h PORT1: Register not supported PORT2: 0000_0000h PORT3: Register not supported

Fields

Field	Function
31-16 —	Reserved
15 LK	Lock Register Locks this PCR. When a PCR*n* is locked, its fields cannot be updated until the next reset. 0b - Does not lock 1b - Locks
14 —	Reserved
13 INV	Invert Input Inverts the digital input. 0b - Does not invert 1b - Inverts
12 IBE	Input Buffer Enable Enables digital input buffer. **NOTE** When a pin is configured as digital peripheral (including GPIO) through PCR[MUX]: • If the peripheral is input or bidirectional function, this bit must be 1. • If the peripheral is output only, the value of this bit doesn't matter. However, if it's 1, you can get the pin state through reading PDIR register.

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

<table>
<tr><th>Field</th><th>Function</th></tr>
<tr><td></td><td>0b - Disables
1b - Enables</td></tr>
<tr><td>11-8
MUX</td><td>Pin Multiplex Control
Configures the multiplexing slots on each pin.
Not all pins support all pin multiplexing slots. Unimplemented pin multiplexing slots are reserved.
Unimplemented pin multiplexing slots can result in different behaviors, if the unimplemented slot is phantomed (because the module is phantomed on that die) versus unimplemented on the die.
The corresponding pin is configured according to the following pin multiplexing slots:
NOTE
This field is not supported in every instance. The following table includes only supported registers.
<table>
<tr><th>Instance</th><th>Field supported in</th><th>Field not supported in</th></tr>
<tr><td>PORT0</td><td>PCR16[10–8]</td><td>PCR16[11]</td></tr>
<tr><td>PORT2</td><td>PCR12–PCR13
PCR16</td><td>—</td></tr>
<tr><td>PORT3</td><td>PCR12–PCR14</td><td>—</td></tr>
</table>
NOTE
The descriptions of the field settings vary by module instance.
<table>
<tr><th>Instance</th><th>Register</th><th>Field value and description</th></tr>
<tr><td>PORT0</td><td>PCR12–PCR14</td><td rowspan="3"></td></tr>
<tr><td>PORT2</td><td>PCR14</td></tr>
<tr><td>PORT3</td><td>PCR16</td></tr>
<tr><td>PORT0</td><td>PCR16</td><td>000b - Alternative 0 (GPIO)
001b - Alternative 1 (chip-specific)
010b - Alternative 2 (chip-specific)
011b - Alternative 3 (chip-specific)
100b - Alternative 4 (chip-specific)
101b - Alternative 5 (chip-specific)
110b - Alternative 6 (chip-specific)</td></tr>
</table>
</td></tr>
</table>

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

<table>
<tr><th>Field</th><th>Function</th></tr>
<tr><td></td><td>
<table>
<tr><th>Instance</th><th>Register</th><th>Field value and description</th></tr>
<tr><td></td><td></td><td>111b - Alternative 7 (chip-specific)</td></tr>
<tr><td rowspan="2">PORT2</td><td>PCR12–PCR13</td><td rowspan="3">0000b - Alternative 0 (GPIO)
0001b - Alternative 1 (chip-specific)
0010b - Alternative 2 (chip-specific)
0011b - Alternative 3 (chip-specific)
0100b - Alternative 4 (chip-specific)
0101b - Alternative 5 (chip-specific)
0110b - Alternative 6 (chip-specific)
0111b - Alternative 7 (chip-specific)
1000b - Alternative 8 (chip-specific)
1001b - Alternative 9 (chip-specific)
1010b - Alternative 10 (chip-specific)
1011b - Alternative 11 (chip-specific)
1100b - Alternative 12 (chip-specific)
1101b - Alternative 13 (chip-specific)</td></tr>
<tr><td>PCR16</td></tr>
<tr><td>PORT3</td><td>PCR12–PCR14</td></tr>
</table>
</td></tr>
<tr><td>7
—</td><td>Reserved</td></tr>
<tr><td>6
DSE</td><td>Drive Strength Enable
Configures drive strength, low or high, on each pin.
The drive strength configuration is valid for all digital pin multiplexing modes.
• When this field = 0, low drive strength is configured on the corresponding pin, if the pin is configured as a digital output.
• When this field = 1, high drive strength is configured on the corresponding pin, if the pin is configured as a digital output.
0b - Low
1b - High</td></tr>
<tr><td>5
ODE</td><td>Open Drain Enable
Enables open drain output on each pin.
The open drain configuration is valid for all digital pin multiplexing modes.
• When this field = 0, the open drain output is disabled on the corresponding pin.</td></tr>
</table>

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	• When this field = 1, the open drain output is enabled on the corresponding pin, if the pin is configured as a digital output. 0b - Disables 1b - Enables
4 —	Reserved
3 SRE	Slew Rate Enable Configures the slew rate feature, fast or slow, on each corresponding pin. The slew rate configuration is valid for all digital pin multiplexing modes. 0b - Fast 1b - Slow
2 —	Reserved
1 PE	Pull Enable Enables the internal pull resistor. This configuration is valid for all digital pin multiplexing modes. • When this field = 0, the internal pull resistor is not enabled on the corresponding pin. • When this field = 1, the internal pull resistor is enabled on the corresponding pin, if the pin is configured as a digital input. 0b - Disables 1b - Enables
0 PS	Pull Select Enables the internal pullup or pulldown resistor. This configuration is valid for all digital pin multiplexing modes. • When this field = 0, the internal pulldown resistor is enabled on the corresponding pin, if the corresponding PCR*n*.PE field = 1. • When this field = 1, the internal pullup resistor is enabled on the corresponding pin, if the corresponding PCR*n*.PE field = 1. 0b - Enables internal pulldown resistor 1b - Enables internal pullup resistor

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

10.6.1.17 Pin Control a (PCR17 - PCR21)

Offset

Register	Offset
PCR17	C4h
PCR20	D0h
PCR21	D4h

Function

Configures pin control features on each pin.

NOTE

Do not modify pin configuration registers associated with pins that are unavailable in your selected package. All unbonded pins unavailable in your package default to the Disabled state for lowest power consumption.

NOTE

Each module instance supports a different number of registers.

Instance	Register supported	Register not supported
PORT0	PCR17	PCR20–PCR21
PORT1	—	PCR17 PCR20–PCR21
PORT2	PCR17 PCR20–PCR21	—
PORT3	—	PCR17 PCR20–PCR21

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	See Register reset values.															

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	LK	0	INV	IBE	0	MUX			0	DSE	ODE	0	SRE	0	PE	PS
W																
Reset	See Register reset values.															

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Register reset values

Register	Reset value
PCR17	PORT0: 0000_0000h PORT1: Register not supported PORT2: 0000_0000h PORT3: Register not supported
PCR18–PCR19	—
PCR20–PCR21	0000_0000h

Fields

Field	Function
31-16 —	Reserved
15 LK	Lock Register Locks this PCR. When a PCR*n* is locked, its fields cannot be updated until the next reset. 0b - Does not lock 1b - Locks
14 —	Reserved
13 INV	Invert Input Inverts the digital input. 0b - Does not invert 1b - Inverts
12 IBE	Input Buffer Enable Enables digital input buffer. **NOTE** When a pin is configured as digital peripheral (including GPIO) through PCR[MUX]: • If the peripheral is input or bidirectional function, this bit must be 1. • If the peripheral is output only, the value of this bit doesn't matter. However, if it's 1, you can get the pin state through reading PDIR register. 0b - Disables 1b - Enables
11	Reserved

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
—	
10-8 MUX	Pin Multiplex Control Configures the multiplexing slots on each pin. Not all pins support all pin multiplexing slots. Unimplemented pin multiplexing slots are reserved. Unimplemented pin multiplexing slots can result in different behaviors, if the unimplemented slot is phantomed (because the module is phantomed on that die) versus unimplemented on the die. The corresponding pin is configured according to the following pin multiplexing slots: 000b - Alternative 0 (GPIO) 001b - Alternative 1 (chip-specific) 010b - Alternative 2 (chip-specific) 011b - Alternative 3 (chip-specific) 100b - Alternative 4 (chip-specific) 101b - Alternative 5 (chip-specific) 110b - Alternative 6 (chip-specific) 111b - Alternative 7 (chip-specific)
7 —	Reserved
6 DSE	Drive Strength Enable Configures drive strength, low or high, on each pin. The drive strength configuration is valid for all digital pin multiplexing modes. • When this field = 0, low drive strength is configured on the corresponding pin, if the pin is configured as a digital output. • When this field = 1, high drive strength is configured on the corresponding pin, if the pin is configured as a digital output. 0b - Low 1b - High
5 ODE	Open Drain Enable Enables open drain output on each pin. The open drain configuration is valid for all digital pin multiplexing modes. • When this field = 0, the open drain output is disabled on the corresponding pin. • When this field = 1, the open drain output is enabled on the corresponding pin, if the pin is configured as a digital output. 0b - Disables

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	1b - Enables
4 —	Reserved
3 SRE	Slew Rate Enable Configures the slew rate feature, fast or slow, on each corresponding pin. The slew rate configuration is valid for all digital pin multiplexing modes. 0b - Fast 1b - Slow
2 —	Reserved
1 PE	Pull Enable Enables the internal pull resistor. This configuration is valid for all digital pin multiplexing modes. • When this field = 0, the internal pull resistor is not enabled on the corresponding pin. • When this field = 1, the internal pull resistor is enabled on the corresponding pin, if the pin is configured as a digital input. 0b - Disables 1b - Enables
0 PS	Pull Select Enables the internal pullup or pulldown resistor. This configuration is valid for all digital pin multiplexing modes. • When this field = 0, the internal pulldown resistor is enabled on the corresponding pin, if the corresponding PCR*n*.PE field = 1. • When this field = 1, the internal pullup resistor is enabled on the corresponding pin, if the corresponding PCR*n*.PE field = 1. 0b - Enables internal pulldown resistor 1b - Enables internal pullup resistor

10.6.1.18 Pin Control 27 (PCR27)

Offset

Register	Offset
PCR27	ECh

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Function

Configures pin control features on each pin.

NOTE

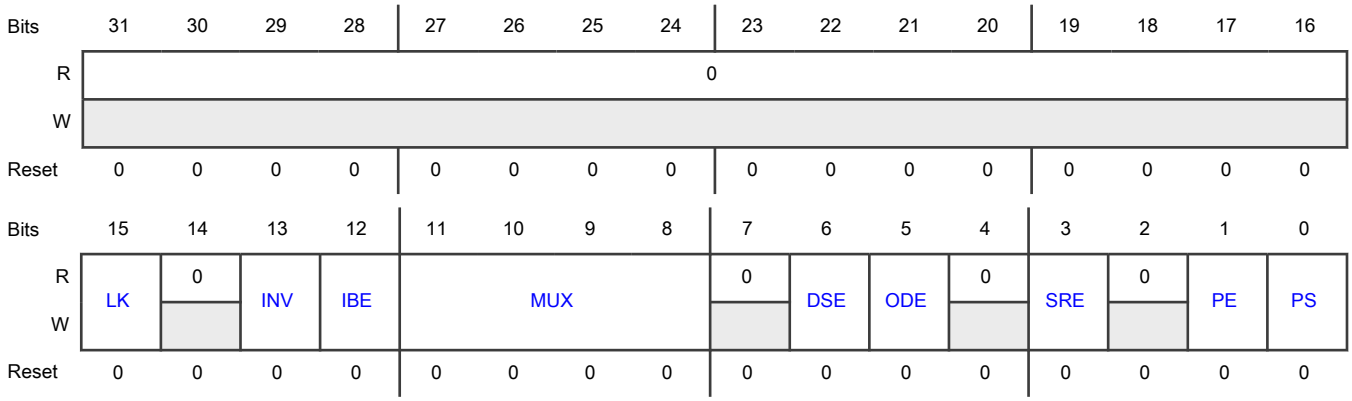
Do not modify pin configuration registers associated with pins that are unavailable in your selected package. All unbonded pins unavailable in your package default to the Disabled state for lowest power consumption.

NOTE

Each module instance supports a different number of registers.

Instance	Register supported	Register not supported
PORT0	—	PCR27
PORT1	—	PCR27
PORT2	PCR27	—
PORT3	PCR27	—

Diagram



Fields

Field	Function
31-16 —	Reserved
15 LK	Lock Register Locks this PCR. When a PCR*n* is locked, its fields cannot be updated until the next reset. 0b - Does not lock 1b - Locks

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

<table>
<tr><th>Field</th><th>Function</th></tr>
<tr><td>14
—</td><td>Reserved</td></tr>
<tr><td>13
INV</td><td>Invert Input
Inverts the digital input.
0b - Does not invert
1b - Inverts</td></tr>
<tr><td>12
IBE</td><td>Input Buffer Enable
Enables digital input buffer.
NOTE
When a pin is configured as digital peripheral (including GPIO) through PCR[MUX]:
• If the peripheral is input or bidirectional function, this bit must be 1.
• If the peripheral is output only, the value of this bit doesn’t matter. However, if it's 1, you can get the pin state through reading PDIR register.
0b - Disables
1b - Enables</td></tr>
<tr><td>11-8
MUX</td><td>Pin Multiplex Control
Configures the multiplexing slots on each pin.
Not all pins support all pin multiplexing slots. Unimplemented pin multiplexing slots are reserved.
Unimplemented pin multiplexing slots can result in different behaviors, if the unimplemented slot is phantomed (because the module is phantomed on that die) versus unimplemented on the die.
The corresponding pin is configured according to the following pin multiplexing slots:
NOTE
This field is not supported in every instance. The following table includes only supported registers.
<table><tr><th>Instance</th><th>Field supported in</th><th>Field not supported in</th></tr><tr><td>PORT2</td><td>PCR27[9–8]</td><td>PCR27[11–10]</td></tr><tr><td>PORT3</td><td>PCR27</td><td>—</td></tr></table>NOTE
The descriptions of the field settings vary by module instance.</td></tr>
</table>

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

<table>
<tr><th>Field</th><th>Function</th></tr>
<tr><td></td><td>
<table>
<tr><th>Instance</th><th>Field value and description</th></tr>
<tr><td>PORT2</td><td>00b - Alternative 0 (GPIO)
01b - Alternative 1 (chip-specific)
10b - Alternative 2 (chip-specific)
11b - Alternative 3 (chip-specific)</td></tr>
<tr><td>PORT3</td><td>0000b - Alternative 0 (GPIO)
0001b - Alternative 1 (chip-specific)
0010b - Alternative 2 (chip-specific)
0011b - Alternative 3 (chip-specific)
0100b - Alternative 4 (chip-specific)
0101b - Alternative 5 (chip-specific)
0110b - Alternative 6 (chip-specific)
0111b - Alternative 7 (chip-specific)
1000b - Alternative 8 (chip-specific)
1001b - Alternative 9 (chip-specific)
1010b - Alternative 10 (chip-specific)
1011b - Alternative 11 (chip-specific)
1100b - Alternative 12 (chip-specific)
1101b - Alternative 13 (chip-specific)</td></tr>
</table>
</td></tr>
<tr><td>7
—</td><td>Reserved</td></tr>
<tr><td>6
DSE</td><td>Drive Strength Enable
Configures drive strength, low or high, on each pin.
The drive strength configuration is valid for all digital pin multiplexing modes.
• When this field = 0, low drive strength is configured on the corresponding pin, if the pin is configured as a digital output.
• When this field = 1, high drive strength is configured on the corresponding pin, if the pin is configured as a digital output.
0b - Low
1b - High</td></tr>
<tr><td>5</td><td>Open Drain Enable</td></tr>
</table>

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
ODE	Enables open drain output on each pin. The open drain configuration is valid for all digital pin multiplexing modes. • When this field = 0, the open drain output is disabled on the corresponding pin. • When this field = 1, the open drain output is enabled on the corresponding pin, if the pin is configured as a digital output. 0b - Disables 1b - Enables
4 —	Reserved
3 SRE	Slew Rate Enable Configures the slew rate feature, fast or slow, on each corresponding pin. The slew rate configuration is valid for all digital pin multiplexing modes. 0b - Fast 1b - Slow
2 —	Reserved
1 PE	Pull Enable Enables the internal pull resistor. This configuration is valid for all digital pin multiplexing modes. • When this field = 0, the internal pull resistor is not enabled on the corresponding pin. • When this field = 1, the internal pull resistor is enabled on the corresponding pin, if the pin is configured as a digital input. 0b - Disables 1b - Enables
0 PS	Pull Select Enables the internal pullup or pulldown resistor. This configuration is valid for all digital pin multiplexing modes. • When this field = 0, the internal pulldown resistor is enabled on the corresponding pin, if the corresponding PCR*n*.PE field = 1. • When this field = 1, the internal pullup resistor is enabled on the corresponding pin, if the corresponding PCR*n*.PE field = 1. 0b - Enables internal pulldown resistor 1b - Enables internal pullup resistor

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

10.6.1.19 Pin Control 28 (PCR28)

Offset

Register	Offset
PCR28	F0h

Function

Configures pin control features on each pin.

NOTE

Do not modify pin configuration registers associated with pins that are unavailable in your selected package. All unbonded pins unavailable in your package default to the Disabled state for lowest power consumption.

NOTE

Each module instance supports a different number of registers.

Instance	Register supported	Register not supported
PORT0	—	PCR28
PORT1	—	PCR28
PORT2	PCR28	—
PORT3	PCR28	—

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	LK	0	INV	IBE	0	MUX			DSE1	DSE	ODE	0	SRE	0	PE	PS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-16 —	Reserved

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
15 LK	Lock Register Locks this PCR. When a PCR*n* is locked, its fields cannot be updated until the next reset. 0b - Does not lock 1b - Locks
14 —	Reserved
13 INV	Invert Input Inverts the digital input. 0b - Does not invert 1b - Inverts
12 IBE	Input Buffer Enable Enables digital input buffer. **NOTE** When a pin is configured as digital peripheral (including GPIO) through PCR[MUX]: • If the peripheral is input or bidirectional function, this bit must be 1. • If the peripheral is output only, the value of this bit doesn't matter. However, if it's 1, you can get the pin state through reading PDIR register. 0b - Disables 1b - Enables
11 —	Reserved
10-8 MUX	Pin Multiplex Control Configures the multiplexing slots on each pin. Not all pins support all pin multiplexing slots. Unimplemented pin multiplexing slots are reserved. Unimplemented pin multiplexing slots can result in different behaviors, if the unimplemented slot is phantomed (because the module is phantomed on that die) versus unimplemented on the die. The corresponding pin is configured according to the following pin multiplexing slots: **NOTE** This field is not supported in every instance. The following table includes only supported registers.

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	<table><tr><th>Instance</th><th>Field supported in</th><th>Field not supported in</th></tr><tr><td>PORT2</td><td>PCR28[9–8]</td><td>PCR28[10]</td></tr><tr><td>PORT3</td><td>PCR28</td><td>—</td></tr></table> **NOTE** The descriptions of the field settings vary by module instance. <table><tr><th>Instance</th><th>Field value and description</th></tr><tr><td>PORT2</td><td>00b - Alternative 0 (GPIO) 01b - Alternative 1 (chip-specific) 10b - Alternative 2 (chip-specific) 11b - Alternative 3 (chip-specific)</td></tr><tr><td>PORT3</td><td>000b - Alternative 0 (GPIO) 001b - Alternative 1 (chip-specific) 010b - Alternative 2 (chip-specific) 011b - Alternative 3 (chip-specific) 100b - Alternative 4 (chip-specific) 101b - Alternative 5 (chip-specific) 110b - Alternative 6 (chip-specific) 111b - Alternative 7 (chip-specific)</td></tr></table>
7 DSE1	Drive Strength Enable Configures the drive strength on each pin. The drive strength configuration is valid for all digital pin multiplexing modes. • If this field = 0, normal drive strength is configured on the corresponding pin. • If this field = 1, double drive strength is configured on the corresponding pin. **NOTE** This field is not supported in every instance. The following table includes only supported registers. <table><tr><th>Instance</th><th>Field supported in</th><th>Field not supported in</th></tr><tr><td>PORT2</td><td>—</td><td>PCR28</td></tr></table>

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	<table><tr><th>Instance</th><th>Field supported in</th><th>Field not supported in</th></tr><tr><td>PORT3</td><td>PCR28</td><td>—</td></tr></table> 0b - Normal 1b - Double
6 DSE	Drive Strength Enable Configures drive strength, low or high, on each pin. The drive strength configuration is valid for all digital pin multiplexing modes. • When this field = 0, low drive strength is configured on the corresponding pin, if the pin is configured as a digital output. • When this field = 1, high drive strength is configured on the corresponding pin, if the pin is configured as a digital output. 0b - Low 1b - High
5 ODE	Open Drain Enable Enables open drain output on each pin. The open drain configuration is valid for all digital pin multiplexing modes. • When this field = 0, the open drain output is disabled on the corresponding pin. • When this field = 1, the open drain output is enabled on the corresponding pin, if the pin is configured as a digital output. 0b - Disables 1b - Enables
4 —	Reserved
3 SRE	Slew Rate Enable Configures the slew rate feature, fast or slow, on each corresponding pin. The slew rate configuration is valid for all digital pin multiplexing modes. 0b - Fast 1b - Slow
2 —	Reserved
1	Pull Enable

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
PE	Enables the internal pull resistor. This configuration is valid for all digital pin multiplexing modes. • When this field = 0, the internal pull resistor is not enabled on the corresponding pin. • When this field = 1, the internal pull resistor is enabled on the corresponding pin, if the pin is configured as a digital input. 0b - Disables 1b - Enables
0 PS	Pull Select Enables the internal pullup or pulldown resistor. This configuration is valid for all digital pin multiplexing modes. • When this field = 0, the internal pulldown resistor is enabled on the corresponding pin, if the corresponding PCR*n*.PE field = 1. • When this field = 1, the internal pullup resistor is enabled on the corresponding pin, if the corresponding PCR*n*.PE field = 1. 0b - Enables internal pulldown resistor 1b - Enables internal pullup resistor

10.6.1.20 Pin Control 29 (PCR29)

Offset

Register	Offset
PCR29	F4h

Function

Configures pin control features on each pin.

NOTE

Do not modify pin configuration registers associated with pins that are unavailable in your selected package. All unbonded pins unavailable in your package default to the Disabled state for lowest power consumption.

NOTE

Each module instance supports a different number of registers.

Instance	Register supported	Register not supported
PORT0	—	PCR29

Table continues on the next page...

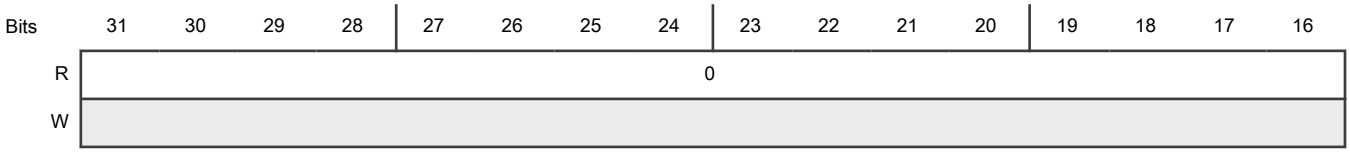
All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Instance	Register supported	Register not supported
PORT1	PCR29	—
PORT2	—	PCR29
PORT3	PCR29	—

Diagram



Register reset values

Register	Reset value
PCR29	PORT0: Register not supported PORT1: 0000_0133h PORT2: Register not supported PORT3: 0000_1103h

Fields

Field	Function
31-16 —	Reserved
15 LK	Lock Register Locks this PCR. When a PCR*n* is locked, its fields cannot be updated until the next reset. 0b - Does not lock 1b - Locks
14	Reserved

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
—	
13 INV	Invert Input Inverts the digital input. 0b - Does not invert 1b - Inverts
12 IBE	Input Buffer Enable Enables digital input buffer. **NOTE** When a pin is configured as digital peripheral (including GPIO) through PCR[MUX]: • If the peripheral is input or bidirectional function, this bit must be 1. • If the peripheral is output only, the value of this bit doesn't matter. However, if it's 1, you can get the pin state through reading PDIR register. 0b - Disables 1b - Enables
11-8 MUX	Pin Multiplex Control Configures the multiplexing slots on each pin. Not all pins support all pin multiplexing slots. Unimplemented pin multiplexing slots are reserved. Unimplemented pin multiplexing slots can result in different behaviors, if the unimplemented slot is phantomed (because the module is phantomed on that die) versus unimplemented on the die. The corresponding pin is configured according to the following pin multiplexing slots: **NOTE** This field is not supported in every instance. The following table includes only supported registers.

Instance	Field supported in	Field not supported in
PORT1	PCR29[9–8]	PCR29[11–10]
PORT3	PCR29	—

NOTE

The descriptions of the field settings vary by module instance.

Instance	Field value and description
PORT1	00b - Alternative 0 (GPIO)

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	<table><tr><th>Instance</th><th>Field value and description</th></tr><tr><td></td><td>01b - Alternative 1 (chip-specific) 10b - Alternative 2 (chip-specific) 11b - Alternative 3 (chip-specific)</td></tr><tr><td>PORT3</td><td>0000b - Alternative 0 (GPIO) 0001b - Alternative 1 (chip-specific) 0010b - Alternative 2 (chip-specific) 0011b - Alternative 3 (chip-specific) 0100b - Alternative 4 (chip-specific) 0101b - Alternative 5 (chip-specific) 0110b - Alternative 6 (chip-specific) 0111b - Alternative 7 (chip-specific) 1000b - Alternative 8 (chip-specific) 1001b - Alternative 9 (chip-specific) 1010b - Alternative 10 (chip-specific) 1011b - Alternative 11 (chip-specific) 1100b - Alternative 12 (chip-specific) 1101b - Alternative 13 (chip-specific)</td></tr></table>
7 — 	Reserved
6 DSE	Drive Strength Enable Configures drive strength, low or high, on each pin. The drive strength configuration is valid for all digital pin multiplexing modes. • When this field = 0, low drive strength is configured on the corresponding pin, if the pin is configured as a digital output. • When this field = 1, high drive strength is configured on the corresponding pin, if the pin is configured as a digital output. 0b - Low 1b - High
5 ODE	Open Drain Enable Enables open drain output on each pin.

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

<table>
<tr><th>Field</th><th>Function</th></tr>
<tr><td></td><td>The open drain configuration is valid for all digital pin multiplexing modes.
• When this field = 0, the open drain output is disabled on the corresponding pin.
• When this field = 1, the open drain output is enabled on the corresponding pin, if the pin is configured as a digital output.
0b - Disables
1b - Enables</td></tr>
<tr><td>4
PFE</td><td>Passive Filter Enable
Enables passive input filter on each pin.
The passive filter configuration is valid for all digital pin multiplexing modes.
• When this field = 0, the passive input filter is disabled on the corresponding pin.
• When this field = 1, the passive input filter is enabled on the corresponding pin, if the pin is configured as a digital input.
See the chip's data sheet for filter characteristics.
NOTE
This field is not supported in every instance. The following table includes only supported registers.
<table><tr><th>Instance</th><th>Field supported in</th><th>Field not supported in</th></tr><tr><td>PORT1</td><td>PCR29</td><td>—</td></tr><tr><td>PORT3</td><td>—</td><td>PCR29</td></tr></table>
0b - Disables
1b - Enables</td></tr>
<tr><td>3
SRE</td><td>Slew Rate Enable
Configures the slew rate feature, fast or slow, on each corresponding pin.
The slew rate configuration is valid for all digital pin multiplexing modes.
0b - Fast
1b - Slow</td></tr>
<tr><td>2
PV</td><td>Pull Value
Selects high or low internal pull resistor value.
The pull value configuration is valid for all digital pin multiplexing modes.</td></tr>
</table>

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	NOTE This field is not supported in every instance. The following table includes only supported registers.

Instance	Field supported in	Field not supported in
PORT1	PCR29	—
PORT3	—	PCR29

Field	Function
	0b - Low 1b - High
1 PE	Pull Enable Enables the internal pull resistor. This configuration is valid for all digital pin multiplexing modes. • When this field = 0, the internal pull resistor is not enabled on the corresponding pin. • When this field = 1, the internal pull resistor is enabled on the corresponding pin, if the pin is configured as a digital input. 0b - Disables 1b - Enables
0 PS	Pull Select Enables the internal pullup or pulldown resistor. This configuration is valid for all digital pin multiplexing modes. • When this field = 0, the internal pulldown resistor is enabled on the corresponding pin, if the corresponding PCR*n*.PE field = 1. • When this field = 1, the internal pullup resistor is enabled on the corresponding pin, if the corresponding PCR*n*.PE field = 1. 0b - Enables internal pulldown resistor 1b - Enables internal pullup resistor

10.6.1.21 Pin Control a (PCR30 - PCR31)

Offset

Register	Offset
PCR30	F8h
PCR31	FCh

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Function

Configures pin control features on each pin.

NOTE

Do not modify pin configuration registers associated with pins that are unavailable in your selected package. All unbonded pins unavailable in your package default to the Disabled state for lowest power consumption.

NOTE

Each module instance supports a different number of registers.

Instance	Register supported	Register not supported
PORT0	—	PCR30–PCR31
PORT1	PCR30–PCR31	—
PORT2	—	PCR30–PCR31
PORT3	PCR30–PCR31	—

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	LK	0	INV	IBE	MUX				0	DSE	ODE	0	SRE	0	PE	PS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-16 —	Reserved
15 LK	Lock Register Locks this PCR. When a PCR*n* is locked, its fields cannot be updated until the next reset. 0b - Does not lock 1b - Locks

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
14 —	Reserved
13 INV	Invert Input Inverts the digital input. 0b - Does not invert 1b - Inverts
12 IBE	Input Buffer Enable Enables digital input buffer. **NOTE** When a pin is configured as digital peripheral (including GPIO) through PCR[MUX]: • If the peripheral is input or bidirectional function, this bit must be 1. • If the peripheral is output only, the value of this bit doesn't matter. However, if it's 1, you can get the pin state through reading PDIR register. 0b - Disables 1b - Enables
11-8 MUX	Pin Multiplex Control Configures the multiplexing slots on each pin. Not all pins support all pin multiplexing slots. Unimplemented pin multiplexing slots are reserved. Unimplemented pin multiplexing slots can result in different behaviors, if the unimplemented slot is phantomed (because the module is phantomed on that die) versus unimplemented on the die. The corresponding pin is configured according to the following pin multiplexing slots: 0000b - Alternative 0 (GPIO) 0001b - Alternative 1 (chip-specific) 0010b - Alternative 2 (chip-specific) 0011b - Alternative 3 (chip-specific) 0100b - Alternative 4 (chip-specific) 0101b - Alternative 5 (chip-specific) 0110b - Alternative 6 (chip-specific) 0111b - Alternative 7 (chip-specific) 1000b - Alternative 8 (chip-specific) 1001b - Alternative 9 (chip-specific) 1010b - Alternative 10 (chip-specific) 1011b - Alternative 11 (chip-specific)

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	1100b - Alternative 12 (chip-specific) 1101b - Alternative 13 (chip-specific)
7 —	Reserved
6 DSE	Drive Strength Enable Configures drive strength, low or high, on each pin. The drive strength configuration is valid for all digital pin multiplexing modes. • When this field = 0, low drive strength is configured on the corresponding pin, if the pin is configured as a digital output. • When this field = 1, high drive strength is configured on the corresponding pin, if the pin is configured as a digital output. 0b - Low 1b - High
5 ODE	Open Drain Enable Enables open drain output on each pin. The open drain configuration is valid for all digital pin multiplexing modes. • When this field = 0, the open drain output is disabled on the corresponding pin. • When this field = 1, the open drain output is enabled on the corresponding pin, if the pin is configured as a digital output. 0b - Disables 1b - Enables
4 —	Reserved
3 SRE	Slew Rate Enable Configures the slew rate feature, fast or slow, on each corresponding pin. The slew rate configuration is valid for all digital pin multiplexing modes. 0b - Fast 1b - Slow
2 —	Reserved
1 PE	Pull Enable Enables the internal pull resistor.

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	This configuration is valid for all digital pin multiplexing modes. • When this field = 0, the internal pull resistor is not enabled on the corresponding pin. • When this field = 1, the internal pull resistor is enabled on the corresponding pin, if the pin is configured as a digital input. 0b - Disables 1b - Enables
0 PS	Pull Select Enables the internal pullup or pulldown resistor. This configuration is valid for all digital pin multiplexing modes. • When this field = 0, the internal pulldown resistor is enabled on the corresponding pin, if the corresponding PCR*n*.PE field = 1. • When this field = 1, the internal pullup resistor is enabled on the corresponding pin, if the corresponding PCR*n*.PE field = 1. 0b - Enables internal pulldown resistor 1b - Enables internal pullup resistor

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Chapter 11
General Purpose Input/Output (GPIO)

11.1 Chip-specific GPIO information

Table 66. Reference links to related information

Topic	Related module	Reference
Full description	GPIO	GPIO
System memory map		Memory map
Clocking		Clock distribution
Power Management		Power Management
Signal multiplexing	Port control	Signal multiplexing

11.1.1 Module instances

This chip has four instances of the GPIO module, GPIO0, GPIO1, GPIO2 and GPIO3.

11.1.2 Interrupt, DMA request, and trigger outputs for GPIO

GPIO modules can be used to trigger interrupts, DMA requests, or trigger outputs (see the INPUTMUX and WUU chapters for details). See the table below for details on the chip-level interrupt, DMA request, and trigger output capability for each GPIO instance.

Table 67. Interrupt, DMA request, and trigger outputs for GPIO

GPIO	Interrupt	DMA	Trigger
GPIO0	Yes	Yes	Yes
GPIO1	Yes	Yes	Yes
GPIO2	Yes	Yes	Yes
GPIO3	Yes	Yes	Yes

11.1.3 GPIO access by I/O port

See I/O port for details about I/O port accessing GPIO.

NOTE

In order to access registers of a GPIO from I/O port, you must configure the related GPIO field in MRCC_GLB_CC2 register to 1.

NOTE

A dummy read of GPIO register must be added before accessing PDIR, PDR or ISFR0 registers.

11.2 Overview

GPIO communicates to the processor core via a zero wait-state interface for maximum pin performance.

All information provided in this document is subject to legal disclaimers.
© 2026 NXP B.V. All rights reserved.

11.2.1 Block diagram

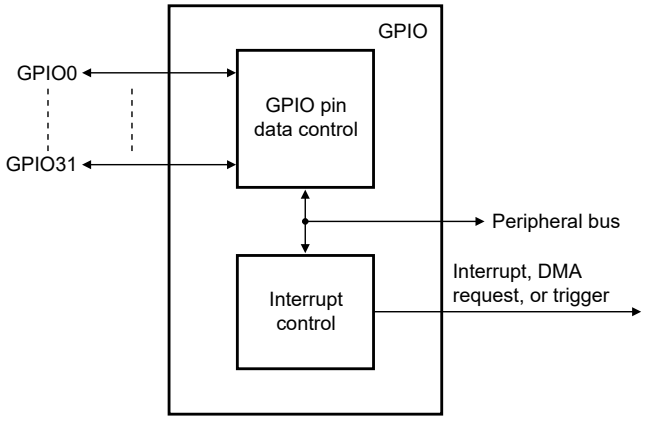


Figure 14. Block diagram

11.2.2 Features

- Port Data Input (PDIR) displays the logic value on each pin when the pin is configured for any digital function, provided the corresponding Port Control and Interrupt modules for that pin are enabled.
- Port Data Output (PDOR), with corresponding set, clear, and toggle registers, controls the output data of each pin when the pin is configured for the GPIO function.
- Port Data Direction (PDDR) controls the direction of each pin when the pin is configured for the GPIO function.
- Port Input Disable (PIDR) controls disabling of the input for each general-purpose pin.
- Pin interrupts:
  - Interrupt flags and enable registers for each pin are functional in all digital pin muxing modes.
  - Support for interrupt, peripheral trigger, or DMA request is configured for each pin.
  - Support for edge-sensitive (rising, falling, or both) or level-sensitive (low, high) interrupts is configured for each pin.
  - Asynchronous wake-up in Low-Power mode.
  - GPIO generates a total of 1 interrupts, 1 output triggers, and 1 DMA requests.

11.3 Functional description

11.3.1 Low-Power mode

You can configure GPIO to exit Low-Power mode via an asynchronous wake-up signal if an enabled interrupt is detected.

11.3.2 Debug mode

GPIO remains functional in Debug mode.

11.3.3 General-purpose input

The logic state of each pin is available via Port Data Input (PDIR) if:

- The corresponding field in Port Input Disable (PIDR) is 0.
- The pin is configured for a digital function.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

11.3.4 General-purpose output

The logic state of each pin is controlled via Port Data Output (PDOR) and Port Data Direction (PDDR), provided the pin is configured for the GPIO function. The following table depicts the conditions for a pin to be configured as input or output.

Table 68. General-purpose output

If	Then
A pin is configured for the GPIO function and the corresponding PDDR field is 0	The pin is configured as an input.
A pin is configured for the GPIO function and the corresponding PDDR field is 1	The pin is configured as an output and the logic state of the pin is equal to the corresponding PDOR field.

For efficient bit manipulation on the general-purpose outputs, pin data set, pin data clear, and pin data toggle registers allow one or more outputs within one port to be set, cleared, or toggled from a single register write, eliminating the need for a read-modify-write operation to prevent changing pins accidentally.

11.3.5 Clocking

GPIO receives a single clock, which is used for register access and synchronization with external pin inputs. There are no special considerations.

11.3.6 Reset

GPIO receives a single reset, which resets the peripheral. There are no special considerations.

11.3.7 External interrupts

The external interrupt capability of GPIO is available in all digital pin muxing modes.

You can configure each pin individually for any of the external interrupt modes shown in the following table.

Table 69. Available pin configurations for external interrupts

Signal conditions	Software polling using flags	Peripheral triggers	Interrupts	DMA requests
Rising-edge	Yes	—	Yes	Yes
Falling-edge	Yes	—	Yes	Yes
Rising- and falling-edge	Yes	—	Yes	Yes
High-level	—	Yes	Yes	—
Low-level	—	Yes	Yes	—

The interrupt status flag is set when the configured edge or level is detected on the pin. Unless GPIO is in Low-Power mode, the input is first synchronized to the system clock to detect the configured level or edge transition.

GPIO generates a pin interrupt that asserts when the interrupt status flag is set for any enabled interrupt for that output. The interrupt negates after the interrupt status flags for all enabled interrupts are cleared by writing a logic 1 to either ISFR0[ISF*n*] or ICR0[ISF].

GPIO generates a DMA request that asserts when the interrupt status flag is set for any enabled DMA request in that output. The DMA request negates after the DMA transfer is completed because that clears the interrupt status flags for all enabled DMA requests.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

In Low-Power mode, the interrupt status flag for any enabled interrupt is asynchronously set if the required level or edge is detected. This also generates an asynchronous wake-up signal to exit Low-Power mode.

GPIO generates a peripheral trigger output that asserts if any pin configured for the active-high trigger is logic 1, or any pin triggered for the active-low trigger is logic 0. The peripheral trigger output asynchronously updates from the value on the configured pins.

11.3.8 Global interrupt control

The two global interrupt control registers (Global Interrupt Control Low (GICLR) and Global Interrupt Control High (GICHR)) allow a single register write to update the upper 16 bits of Interrupt Control a (ICR0 - ICR31) for up to 16 pins, all with the same value.

These global interrupt control registers allow you to quickly configure multiple pins within the same port with the same interrupt configuration.

The global interrupt control registers are write-only registers and always read 0.

11.3.9 DMA

GPIO generates 1 requests. For each pin, you can configure DMA requests to assert for either rising, falling, or both edge detection. See External interrupts for more information.

11.4 External signals

Table 70. External signals

Signal	Description		Direction
GPIO31–GPIO0	General-purpose input/output		I/O
	State meaning	Asserted: The pin is logic 1. Deasserted: The pin is logic 0.	
	Timing	Assertion: When output, this signal occurs on the rising-edge of the system clock. For input, it may occur at any time and input may be asserted asynchronously to the system clock. Deassertion: When output, this signal occurs on the rising-edge of the system clock. For input, it may occur at any time and input may be asserted asynchronously to the system clock.	

NOTE

Not all pins within each GPIO are implemented on each chip. See the "Signal Multiplexing" chapter for the number of GPIO pins within each port available on this chip.

11.5 Initialization

To initialize GPIO:

1. Initialize the GPIO pins for the output function:
   a. Configure the output logic value for each pin by using Port Data Output (PDOR).
   b. Configure the direction for each pin by using Port Data Direction (PDDR).
2. Initialize the interrupt function by writing to Interrupt Control a (ICR0 - ICR31) for the corresponding pins and desired configuration. If the pin is previously used for a different function, first write 0100_0000h to Interrupt Control a (ICR0 - ICR31) to disable the previous function and clear the flag.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

11.6 Application information

GPIO includes the following applications:

- Reading the state of a single pin by performing a byte read of Pin Data (P0DR - P31DR).
- Updating the state of a single pin by performing a byte write to Pin Data (P0DR - P31DR).
- Reading the state of multiple pins by reading Port Data Input (PDIR).
- Updating the state of multiple pins by using the following ways:
  - Writing to Port Data Output (PDOR).
  - Writing to Port Set Output (PSOR) to write 1 to Port Data Output (PDOR).
  - Writing to Port Clear Output (PCOR) to write 0 to Port Data Output (PDOR).
  - Writing to Port Toggle Output (PTOR) to toggle Port Data Output (PDOR).

11.7 Memory map and register definition

The GPIO registers support 8-bit, 16-bit, or 32-bit accesses. Any read or write access to the GPIO memory space that is outside the valid memory map results in a bus error.

NOTE

For simplicity, each GPIO port's register appears with the same width of 32 bits, corresponding to 32 pins. The actual number of pins per port (and therefore the number of usable control bits per port register) is chip-specific. See the "Signal Multiplexing" chapter for the exact control bits of each port.

11.7.1 GPIO register descriptions

11.7.1.1 GPIO memory map

GPIO0 base address: 400F_3000h

GPIO1 base address: 400F_4000h

GPIO2 base address: 400F_5000h

GPIO3 base address: 400F_6000h

Offset	Register	Width (In bits)	Access	Reset value
0h	Version ID (VERID)	32	R	0201_0000h
4h	Parameter (PARAM)	32	R	0000_0001h
40h	Port Data Output (PDOR)	32	RW	0000_0000h
44h	Port Set Output (PSOR)	32	RW	0000_0000h
48h	Port Clear Output (PCOR)	32	RW	0000_0000h
4Ch	Port Toggle Output (PTOR)	32	RW	0000_0000h
50h	Port Data Input (PDIR)	32	R	0000_0000h
54h	Port Data Direction (PDDR)	32	RW	0000_0000h
58h	Port Input Disable (PIDR)	32	RW	0000_0000h

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Offset	Register	Width (In bits)	Access	Reset value
60h - 7Fh	Pin Data (P0DR - P31DR)	8	RW	00h
80h - FCh	Interrupt Control a (ICR0 - ICR31)	32	RW	0000_0000h
100h	Global Interrupt Control Low (GICLR)	32	RW	0000_0000h
104h	Global Interrupt Control High (GICHR)	32	RW	0000_0000h
120h	Interrupt Status Flag (ISFR0)	32	RW	0000_0000h

11.7.1.2 Version ID (VERID)

Offset

Register	Offset
VERID	0h

Function

Indicates the version ID number of each GPIO.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	MAJOR								MINOR							
W																
Reset	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	FEATURE															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-24 MAJOR	Major Version Number Indicates the major version number for the specification.
23-16 MINOR	Minor Version Number Indicates the minor version number for the specification.

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
15-0 FEATURE	Feature Specification Number Indicates the feature set number. 0000_0000_0000_0000b - Basic implementation 0000_0000_0000_0001b - Protection registers implemented

11.7.1.3 Parameter (PARAM)

Offset

Register	Offset
PARAM	4h

Function

Indicates the interrupt number of each GPIO.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0												IRQNUM			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Fields

Field	Function
31-4 —	Reserved
3-0 IRQNUM	Interrupt Number Indicates the number of interrupt, trigger, or DMA request domains.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

11.7.1.4 Port Data Output (PDOR)

Offset

Register	Offset
PDOR	40h

Function

Configures the logic levels that are driven on each general-purpose output pin.

NOTE

Do not modify the pin configuration registers associated with pins that are not available in your selected package. By default, these unbonded pins are set to the Disable state for lowest power consumption.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	PDO31	PDO30	PDO29	PDO28	PDO27	PDO26	PDO25	PDO24	PDO23	PDO22	PDO21	PDO20	PDO19	PDO18	PDO17	PDO16
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PDO15	PDO14	PDO13	PDO12	PDO11	PDO10	PDO9	PDO8	PDO7	PDO6	PDO5	PDO4	PDO3	PDO2	PDO1	PDO0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-0 PDOn	Port Data Output Configures the logic level on the pin if it is configured for general-purpose output. If this field is 0, logic level 0 is driven on the pin, if the pin is configured for general-purpose output. If this field is 1, logic level 1 is driven on the pin, if the pin is configured for general-purpose output. **NOTE** Reading the fields for unbonded pins returns an undefined value. 0b - Logic level 0 1b - Logic level 1

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

11.7.1.5 Port Set Output (PSOR)

Offset

Register	Offset
PSOR	44h

Function

Updates the corresponding fields of Port Data Output (PDOR) to become 1.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
W	PTSO 31	PTSO 30	PTSO 29	PTSO 28	PTSO 27	PTSO 26	PTSO 25	PTSO 24	PTSO 23	PTSO 22	PTSO 21	PTSO 20	PTSO 19	PTSO 18	PTSO 17	PTSO 16
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
W	PTSO 15	PTSO 14	PTSO 13	PTSO 12	PTSO 11	PTSO 10	PTSO 9	PTSO 8	PTSO 7	PTSO 6	PTSO 5	PTSO 4	PTSO 3	PTSO 2	PTSO 1	PTSO 0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-0 PTSOn	Port Set Output Updates the content of the corresponding field in Port Data Output (PDOR). If this field is 0, the corresponding PDOR field does not change. If this field is 1, the corresponding PDOR field becomes 1. 0b - No change 1b - Corresponding field in PDOR becomes 1

11.7.1.6 Port Clear Output (PCOR)

Offset

Register	Offset
PCOR	48h

Function

Updates the corresponding fields of Port Data Output (PDOR) to become 0.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
W	PTCO 31	PTCO 30	PTCO 29	PTCO 28	PTCO 27	PTCO 26	PTCO 25	PTCO 24	PTCO 23	PTCO 22	PTCO 21	PTCO 20	PTCO 19	PTCO 18	PTCO 17	PTCO 16
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
W	PTCO 15	PTCO 14	PTCO 13	PTCO 12	PTCO 11	PTCO 10	PTCO 9	PTCO 8	PTCO 7	PTCO 6	PTCO 5	PTCO 4	PTCO 3	PTCO 2	PTCO 1	PTCO 0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-0 PTCOn	Port Clear Output Updates the content of the corresponding field in Port Data Output (PDOR). If this field is 0, the corresponding PDOR field does not change. If this field is 1, the corresponding PDOR field becomes 0. 0b - No change 1b - Corresponding field in PDOR becomes 0

11.7.1.7 Port Toggle Output (PTOR)

Offset

Register	Offset
PTOR	4Ch

Function

Updates the corresponding fields of Port Data Output (PDOR) to set to the inverse of their current logic states.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
W	PTTO 31	PTTO 30	PTTO 29	PTTO 28	PTTO 27	PTTO 26	PTTO 25	PTTO 24	PTTO 23	PTTO 22	PTTO 21	PTTO 20	PTTO 19	PTTO 18	PTTO 17	PTTO 16
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
W	PTTO 15	PTTO 14	PTTO 13	PTTO 12	PTTO 11	PTTO 10	PTTO 9	PTTO 8	PTTO 7	PTTO 6	PTTO 5	PTTO 4	PTTO 3	PTTO 2	PTTO 1	PTTO 0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-0 PTTOn	Port Toggle Output Updates the content of the corresponding field in Port Data Output (PDOR). If this field is 0, the corresponding PDOR field does not change. If this field is 1, the corresponding PDOR field is set to the inverse of its current logic state. 0b - No change 1b - Set to the inverse of its current logic state

11.7.1.8 Port Data Input (PDIR)

Offset

Register	Offset
PDIR	50h

Function

Captures the logic levels of each general-purpose input pin.

NOTE

Do not modify the pin configuration registers associated with the pins that are not available in your selected package. By default, these unbonded pins are set to the Disable state for lowest power consumption.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	PDI31	PDI30	PDI29	PDI28	PDI27	PDI26	PDI25	PDI24	PDI23	PDI22	PDI21	PDI20	PDI19	PDI18	PDI17	PDI16
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PDI15	PDI14	PDI13	PDI12	PDI11	PDI10	PDI9	PDI8	PDI7	PDI6	PDI5	PDI4	PDI3	PDI2	PDI1	PDI0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-0 PDIn	Port Data Input Indicates the logic level of the pin that is configured for use by a digital function. If this field is 0, the pin logic level is logic 0 or is not configured or implemented for use by a digital function. If this field is 1, the pin logic level is logic 1. 0b - Logic 0 1b - Logic 1

11.7.1.9 Port Data Direction (PDDR)

Offset

Register	Offset
PDDR	54h

Function

Configures the individual port pins for input or output.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	PDD31	PDD30	PDD29	PDD28	PDD27	PDD26	PDD25	PDD24	PDD23	PDD22	PDD21	PDD20	PDD19	PDD18	PDD17	PDD16
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PDD15	PDD14	PDD13	PDD12	PDD11	PDD10	PDD9	PDD8	PDD7	PDD6	PDD5	PDD4	PDD3	PDD2	PDD1	PDD0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Fields

Field	Function
31-0 PDDn	Port Data Direction Configures individual port pins for input or output. If this field is 0, the pin is configured as general-purpose input for the GPIO function. If this field is 1, the pin is configured as general-purpose output for the GPIO function. 0b - Input 1b - Output

11.7.1.10 Port Input Disable (PIDR)

Offset

Register	Offset
PIDR	58h

Function

Disables the input for each general-purpose pin, which prevents the value from being reported in Port Data Input (PDIR).

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	PID31	PID30	PID29	PID28	PID27	PID26	PID25	PID24	PID23	PID22	PID21	PID20	PID19	PID18	PID17	PID16
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PID15	PID14	PID13	PID12	PID11	PID10	PID9	PID8	PID7	PID6	PID5	PID4	PID3	PID2	PID1	PID0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-0 PIDn	Port Input Disable Disables a pin for general-purpose input. If this field is 0, the pin is configured for general-purpose input, provided the pin is configured for a digital function. If this field is 1, the pin is disabled for general-purpose input. 0b - Configured for general-purpose input 1b - Disabled for general-purpose input

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

11.7.1.11 Pin Data (P0DR - P31DR)

Offset

For a = 0 to 31:

Register	Offset
PaDR	60h + (a × 1h)

Function

Configures the data feature of a pin. Pins that are unimplemented or not configured for a digital function read zero.

NOTE

You must not modify Pin Data (P0DR - P31DR) associated with the pins that are not available in your selected package. These unbonded pins are, by default, set to the Disable state for lowest power consumption.

Diagram

Bits	7	6	5	4	3	2	1	0
R	0							PD
W								
Reset	0	0	0	0	0	0	0	0

Fields

Field	Function
7-1 —	Reserved
0 PD	Pin Data (I/O) Specifies the pin logic level. This field updates the corresponding field in Port Data Output (PDOR); reading this field returns the value in the corresponding field of Port Data Input (PDIR). If this field is 0, the pin logic level is logic zero or not configured for use by a digital function. If this field is 1, the pin logic level is logic one. 0b - Logic zero 1b - Logic one

11.7.1.12 Interrupt Control a (ICR0 - ICR31)

Offset

For a = 0 to 31:

Register	Offset
ICRa	80h + (a × 4h)

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Function

Configures interrupt features on each pin.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0							ISF	0	0			IRQC			
W								W1C								
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-25 —	Reserved
24 ISF	Interrupt Status Flag Indicates whether the configured interrupt is detected. The pin interrupt configuration is valid in all digital pin muxing modes. The fields in Interrupt Status Flag (ISFR0) have the same function. ISF can be cleared with either register field. If the pin is configured to generate a DMA request, then the corresponding flag is cleared automatically at the completion of the requested DMA transfer. Otherwise, the flag remains set until a logic 1 is written to the flag. If the pin is configured for a level-sensitive interrupt and the pin remains asserted, then the flag is set again immediately after it is cleared. **NOTE** This field behaves differently for register reads and writes. When reading 0b - Not detected 1b - Detected When writing 0b - No effect 1b - Clear the flag
23 —	Reserved

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
22-20 —	Reserved
19-16 IRQC	Interrupt Configuration Specifies the ISF and DMA request configuration. The pin interrupt configuration is valid in all digital pin muxing modes. When changing the interrupt configuration, it is recommended to first disable ISF and then write the new configuration. The corresponding pin is configured to generate interrupt, trigger, or DMA request. **NOTE** See the GPIO chip-specific information to determine the GPIO instances that support interrupt, DMA request, or trigger capabilities. 0000b - ISF is disabled 0001b - ISF and DMA request on rising edge 0010b - ISF and DMA request on falling edge 0011b - ISF and DMA request on either edge 0100b - Reserved 0101b - ISF sets on rising edge 0110b - ISF sets on falling edge 0111b - ISF sets on either edge 1000b - ISF and interrupt when logic 0 1001b - ISF and interrupt on rising edge 1010b - ISF and interrupt on falling edge 1011b - ISF and Interrupt on either edge 1100b - ISF and interrupt when logic 1 1101b - Enable active-high trigger output; ISF on rising edge (pin state is ORed with other enabled triggers to generate the output trigger for use by other peripherals) 1110b - Enable active-low trigger output; ISF on falling edge (pin state is inverted and ORed with other enabled triggers to generate the output trigger for use by other peripherals) 1111b - Reserved
15-0 —	Reserved

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

11.7.1.13 Global Interrupt Control Low (GICLR)

Offset

Register	Offset
GICLR	100h

Function

Updates any combination of the lower 16 Interrupt Control registers with the same value. This register supports only 32-bit writes and ignores any 16-bit or 8-bit writes.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W	GIWD															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
W	GIWE 15	GIWE 14	GIWE 13	GIWE 12	GIWE 11	GIWE 10	GIWE 9	GIWE 8	GIWE 7	GIWE 6	GIWE 5	GIWE 4	GIWE 3	GIWE 2	GIWE 1	GIWE 0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-16 GIWD	Global Interrupt Write Data Indicates the write value that is written to the upper 16 bits of Interrupt Control a (ICR0 - ICR31), selected by GIWE.
15-0 GIWEn	Global Interrupt Write Enable Indicates whether the upper 16 bits of the corresponding Interrupt Control a (ICR0 - ICR31) are updated with the value in GIWD. 0b - Not updated 1b - Updated

11.7.1.14 Global Interrupt Control High (GICHR)

Offset

Register	Offset
GICHR	104h

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Function

Updates any combination of the upper 16 Interrupt Control registers with the same value. This register supports only 32-bit writes and ignores any 16-bit or 8-bit writes.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W	GIWD															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
W	GIWE 31	GIWE 30	GIWE 29	GIWE 28	GIWE 27	GIWE 26	GIWE 25	GIWE 24	GIWE 23	GIWE 22	GIWE 21	GIWE 20	GIWE 19	GIWE 18	GIWE 17	GIWE 16
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-16 GIWD	Global Interrupt Write Data Indicates the write value that is written to the upper 16 bits of Interrupt Control a (ICR0 - ICR31), selected by GIWE.
15-0 GIWEn	Global Interrupt Write Enable Indicates whether the upper 16 bits of the corresponding Interrupt Control a (ICR0 - ICR31) are updated with the value in GIWD. 0b - Not updated. 1b - Updated

11.7.1.15 Interrupt Status Flag (ISFR0)

Offset

Register	Offset
ISFR0	120h

Function

Indicates whether the related configured interrupt is detected on each pin. The pin interrupt configuration is valid in all digital pin muxing modes. The ISF for each pin is also visible in the corresponding Interrupt Control register, and each flag can be cleared in either location.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	ISF31	ISF30	ISF29	ISF28	ISF27	ISF26	ISF25	ISF24	ISF23	ISF22	ISF21	ISF20	ISF19	ISF18	ISF17	ISF16
W	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	ISF15	ISF14	ISF13	ISF12	ISF11	ISF10	ISF9	ISF8	ISF7	ISF6	ISF5	ISF4	ISF3	ISF2	ISF1	ISF0
W	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-0 ISFn	Interrupt Status Flag Indicates the detection of the configured interrupt on each pin of the same number. If this field is 0, the configured interrupt is not detected on the pin of the same number. If this field is 1, the configured interrupt is detected on the pin of the same number. If the pin is configured to generate a DMA request, then the corresponding flag is cleared automatically at the completion of the requested DMA transfer. Otherwise, the flag remains set until a logic 1 is written to the flag. If the pin is configured for a level-sensitive interrupt and the pin remains asserted, then the flag is set again immediately after it is cleared. **NOTE** This field behaves differently for register reads and writes. When reading 0b - Not detected 1b - Detected When writing 0b - No effect 1b - Clear the flag

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Chapter 12
Input Multiplexing (INPUTMUX)

12.1 Chip-specific INPUTMUX information

Table 71. Reference links to related information

Topic	Related module	Reference
Full description	INPUTMUX	INPUTMUX
System memory map		Memory map
Clocking		Clock distribution
Power Management		Power Management
Signal multiplexing	Port control	Signal multiplexing

Once set up, no clocks are required for the INPUTMUX to function. The system clock is needed only to write to, or read from the INPUTMUX registers. Once the INPUTMUX is configured, disable the clock to the INPUTMUX module in registers of the SYSCON module.

12.1.1 Module instances

This chip contains one instance of the INPUTMUX module.

12.2 Overview

INPUTMUX provides signal routing options for internal peripherals. Some peripheral inputs are multiplexed to multiple input sources. The sources can be external pins, interrupts, output signals of other peripherals, or other internal signals.

NOTE

Depending on the package, not all inputs from external pins are available.

12.2.1 Block diagram

Figure 15 shows a generic input multiplexer arrangement with n inputs.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

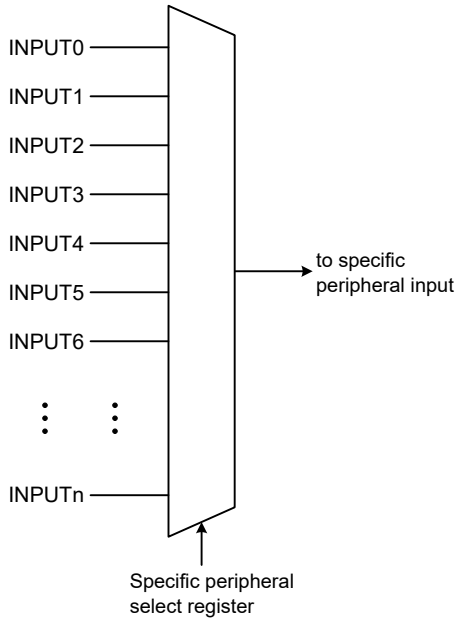


Figure 15. Generic input multiplexing

12.2.2 Features

- Configures the inputs to the asynchronous CTIMERs.
- Configures the inputs to the GPIO event Interrupt.
- Configures the inputs to CMP, ADC, eFlexPWM, eFlexPWM external clock, external trigger, LPI2C, LPSPI and LPUART.
- Configures the inputs to the Frequency Measurement (FREQME). This function is controlled by the FREQME control register.

12.3 TRIG_IN Filter

There is a filter implemented for each TRIG_IN pin. The clock source of the filter is same with input mux bus clock. There is a pre-scaler for the clock. The level of the TRIG_IN can be read from TRIGFIL_STAT register. The filter width is configurable in TRIGFILx_CTRL register. The filtered TRIG_IN is connected to other peripherals though the input mux of the peripheral.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

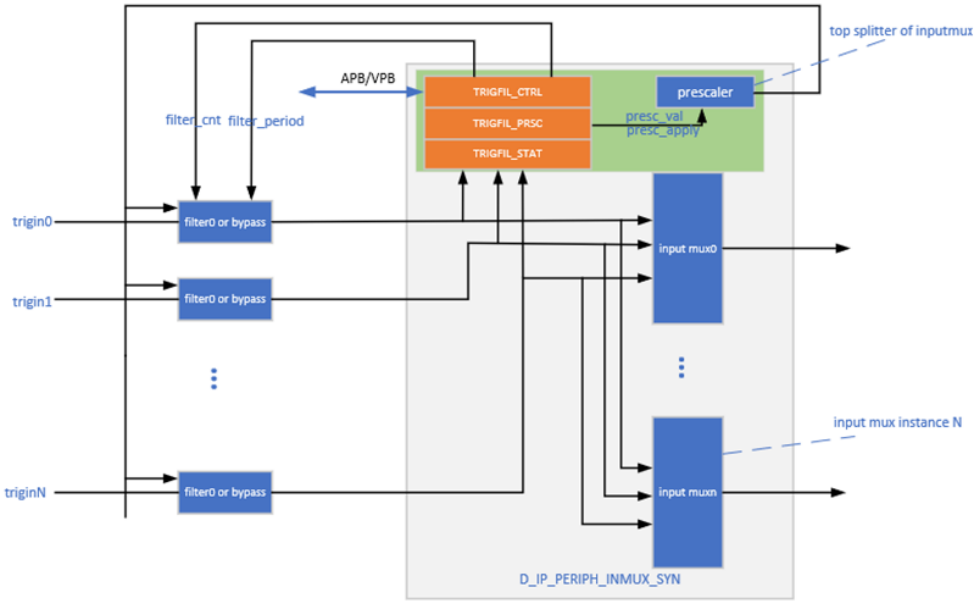


Figure 16. TRIG_IN Filter

12.4 Functional description

The INPUTMUX implements a number of input multiplexers that select one of many inputs to be routed to a specific input signal for a given peripheral. This is used to allow user configuration of data paths between internal modules and/or external pins on the device. For every module input (output from the INPUTMUX), there is a register that selects the input to use, where the register name and description provide details on the module input controlled by each register. The input signal/pin options for each of the muxes are configurable, and can vary from mux to mux. Refer to the register descriptions for the details on the input signal/pin options used for each INPUTMUX output in Memory map and register definition.

12.5 External signals

The INPUTMUX has no dedicated pins. Multiplexer inputs from external pins work independently of any other function assigned to the pin as long as no analog function is enabled.

Table 72. INPUTMUX pin description

Pins	Peripheral	Section
FREQME_xxx[1]	Frequency measure module	See INPUTMUX.
CT_INP[0:6,8:9,12:13,16:19] pin functions selected from PORT register	CTIMER	See CTIMER.
TRIG_IN[0:11] pin functions selected from PORT register	INPUTMUX trigger input	See INPUTMUX.

1. Refer to the pinout sheet attached for more details.

12.6 Memory map and register definition

This section includes the INPUTMUX module memory map and detailed descriptions of all registers.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

12.6.1 INPUTMUX register descriptions

12.6.1.1 INPUTMUX memory map

INPUTMUX0 base address: 4000_1000h

Offset	Register	Width (In bits)	Access	Reset value
20h - 2Ch	Capture Select for CTIMER0 Inputs (CTIMER0CAP0 - CTIMER0CAP3)	32	RW	0000_007Fh
30h	Trigger for CTIMER0 (CTIMER0TRIG)	32	RW	0000_007Fh
40h - 4Ch	Capture Select for CTIMER1 Inputs (CTIMER1CAP0 - CTIMER1CAP3)	32	RW	0000_007Fh
50h	Trigger for CTIMER1 (CTIMER1TRIG)	32	RW	0000_007Fh
180h	FREQME Reference Clock Selection (FREQMEAS_REF)	32	RW	0000_003Fh
184h	FREQME Target Clock Selection (FREQMEAS_TAR)	32	RW	0000_003Fh
260h	LPCMP0 Input Connections (CMP0_TRIG)	32	RW	0000_001Fh
280h - 28Ch	ADC0 Trigger Input Connections (ADC0_TRIG0 - ADC0_TRIG3)	32	RW	0000_001Fh
3A0h	eFlexPWM0 SM0 EXTA Input Trigger Connections (FlexPWM0_SM0_EXTA)	32	RW	0000_003Fh
3A4h	eFlexPWM0 SM0 EXT_SYNC Input Trigger Connections (FlexPWM0_SM0_EXTSYNC)	32	RW	0000_003Fh
3A8h	eFlexPWM0 SM1 EXTA Input Trigger Connections (FlexPWM0_SM1_EXTA)	32	RW	0000_003Fh
3ACh	eFlexPWM0 SM1 EXT_SYNC Input Trigger Connections (FlexPWM0_SM1_EXTSYNC)	32	RW	0000_003Fh
3B0h	eFlexPWM0 SM2 EXTA Input Trigger Connections (FlexPWM0_SM2_EXTA)	32	RW	0000_003Fh
3B4h	eFlexPWM0 SM2 EXT_SYNC Input Trigger Connections (FlexPWM0_SM2_EXTSYNC)	32	RW	0000_003Fh
3C0h	eFlexPWM0 FAULT0 Input Trigger Connections (FlexPWM0_FAULT0)	32	RW	0000_003Fh
3C4h	eFlexPWM0 FAULT1 Input Trigger Connections (FlexPWM0_FAULT1)	32	RW	0000_003Fh
3C8h	eFlexPWM0 FAULT2 Input Trigger Connection (FlexPWM0_FAULT2)	32	RW	0000_003Fh
3CCh	eFlexPWM0 FAULT3 Input Trigger Connection (FlexPWM0_FAULT3)	32	RW	0000_003Fh
3D0h	eFlexPWM0 EXT_FORCE Input Trigger Connection (FlexPWM0_FORCE)	32	RW	0000_003Fh
420h	eFlexPWM0 EXT_CLK Trigger (PWM0_EXT_CLK)	32	RW	0000_000Fh

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Offset	Register	Width (In bits)	Access	Reset value
4C0h	External Trigger a Connections (EXT_TRIG0)	32	RW	0000_001Fh
4C4h	External Trigger a Connections (EXT_TRIG1)	32	RW	0000_001Fh
4C8h	External Trigger a Connections (EXT_TRIG2)	32	RW	0000_001Fh
4CCh	External Trigger a Connections (EXT_TRIG3)	32	RW	0000_001Fh
4D0h	External Trigger a Connections (EXT_TRIG4)	32	RW	0000_001Fh
4D8h	External Trigger a Connections (EXT_TRIG6)	32	RW	0000_001Fh
4DCh	External Trigger a Connections (EXT_TRIG7)	32	RW	0000_001Fh
5A0h	LPI2C0 Trigger Input Connections (LPI2C0_TRIG)	32	RW	0000_001Fh
5E0h	LPSPI0 Trigger Input Connections (LPSPI0_TRIG)	32	RW	0000_001Fh
620h	LPUART0 Trigger Input Connections (LPUART0)	32	RW	0000_003Fh
640h	LPUART1 Trigger Input Connections (LPUART1)	32	RW	0000_003Fh
660h	LPUART2 Trigger Input Connections (LPUART2)	32	RW	0000_003Fh
680h	LPUART3 Trigger Input Connections (LPUART3)	32	RW	0000_003Fh
A00h	Trigger Filter Prescaller (TRIGFIL_PRSC)	32	RW	0000_0000h
A04h	Trigger Filter Status (TRIGFIL_STAT0)	32	R	0000_0000h
A10h - A3Ch	TRIGFIL Control (TRIGFIL0 - TRIGFIL11)	32	RW	0000_0000h

12.6.1.2 Capture Select for CTIMER0 Inputs (CTIMER0CAP0 - CTIMER0CAP3)

Offset

Register	Offset
CTIMER0CAP0	20h
CTIMER0CAP1	24h
CTIMER0CAP2	28h
CTIMER0CAP3	2Ch

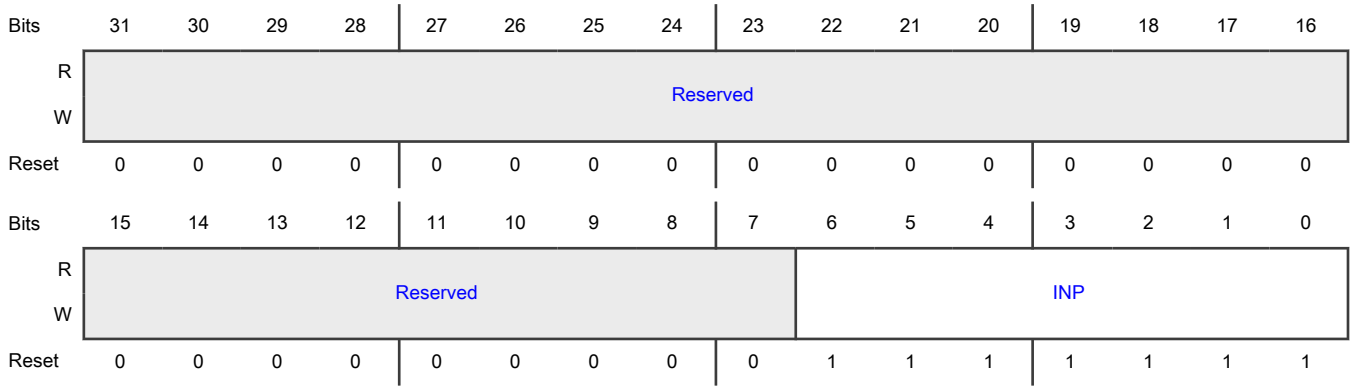
Function

For CTIMER0, there are 4 CTIMER0CAPa, with a = 0 to 3, each allowing selecting between external or internal input sources. The output of CTIMER0CAP0 register selects the source for CTIMER0 capture input 0. The output of CTIMER0CAP1 register selects the source for CTIMER0 capture input 1 and so forth up to CTIMER0CAP3 register, which selects the input for CTIMER0 capture input 3.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram



Fields

Field	Function
31-7 —	Reserved
6-0 INP	Capture Input a for CTIMER0 000_0000b - Reserved 000_0001b - CT_INP0 input is selected 000_0010b - CT_INP1 input is selected 000_0011b - CT_INP2 input is selected 000_0100b - CT_INP3 input is selected 000_0101b - CT_INP4 input is selected 000_0110b - CT_INP5 input is selected 000_0111b - CT_INP6 input is selected 000_1000b - Reserved 000_1001b - CT_INP8 input is selected 000_1010b - CT_INP9 input is selected 000_1011b - Reserved 000_1100b - Reserved 000_1101b - CT_INP12 input is selected 000_1110b - CT_INP13 input is selected 000_1111b - Reserved 001_0000b - Reserved 001_0001b - CT_INP16 input is selected 001_0010b - CT_INP17 input is selected 001_0011b - CT_INP18 input is selected

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	001_0100b - CT_INP19 input is selected 001_0101b - Reserved 001_0110b - Reserved 001_0111b - Reserved 001_1000b - Reserved 001_1001b - Reserved 001_1010b - ADC0_tcomp[0] input is selected 001_1011b - ADC0_tcomp[1] input is selected 001_1100b - ADC0_tcomp[2] input is selected 001_1101b - ADC0_tcomp[3] input is selected 001_1110b - CMP0_OUT input is selected 001_1111b - Reserved 010_0000b - Reserved 010_0001b - CTimer1_MAT1 input is selected 010_0010b - CTimer1_MAT2 input is selected 010_0011b - CTimer1_MAT3 input is selected 010_0100b - Reserved 010_0101b - Reserved 010_0110b - Reserved 010_0111b - Reserved 010_1000b - Reserved 010_1001b - Reserved 010_1010b - Reserved 010_1011b - Reserved 010_1100b - PWM0_SM0_MUX_TRIG0 input is selected 010_1101b - PWM0_SM1_MUX_TRIG0 input is selected 010_1110b - PWM0_SM2_MUX_TRIG0 input is selected 010_1111b - Reserved 011_0000b - LPI2C0 Master End of Packet input is selected 011_0001b - LPI2C0 Slave End of Packet input is selected 011_0010b - Reserved 011_0011b - Reserved 011_0100b - LPSPI0 End of Frame input is selected

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	011_0101b - LPSPI0 Received Data Word input is selected 011_0110b - Reserved 011_0111b - Reserved 011_1000b - LPUART0 Received Data Word input is selected 011_1001b - LPUART0 Transmitted Data Word input is selected 011_1010b - LPUART0 Receive Line Idle input is selected 011_1011b - LPUART1 Received Data Word input is selected 011_1100b - LPUART1 Transmitted Data Word input is selected 011_1101b - LPUART1 Receive Line Idle input is selected 011_1110b - LPUART2 Received Data Word input is selected 011_1111b - LPUART2 Transmitted Data Word input is selected 100_0000b - LPUART2 Receive Line Idle input is selected 100_0001b - TRIG_IN0 input is selected 100_0010b - TRIG_IN1 input is selected 100_0011b - TRIG_IN2 input is selected 100_0100b - TRIG_IN3 input is selected 100_0101b - TRIG_IN4 input is selected 100_0110b - TRIG_IN5 input is selected 100_0111b - TRIG_IN6 input is selected 100_1000b - TRIG_IN7 input is selected 100_1001b - TRIG_IN8 input is selected 100_1010b - TRIG_IN9 input is selected 100_1011b - TRIG_IN10 input is selected 100_1100b - TRIG_IN11 input is selected 100_1101b - LPUART3 Received Data Word input is selected 100_1110b - LPUART3 Transmitted Data Word input is selected 100_1111b - LPUART3 Receive Line Idle input is selected All other values are reserved.

12.6.1.3 Trigger for CTIMER0 (CTIMER0TRIG)

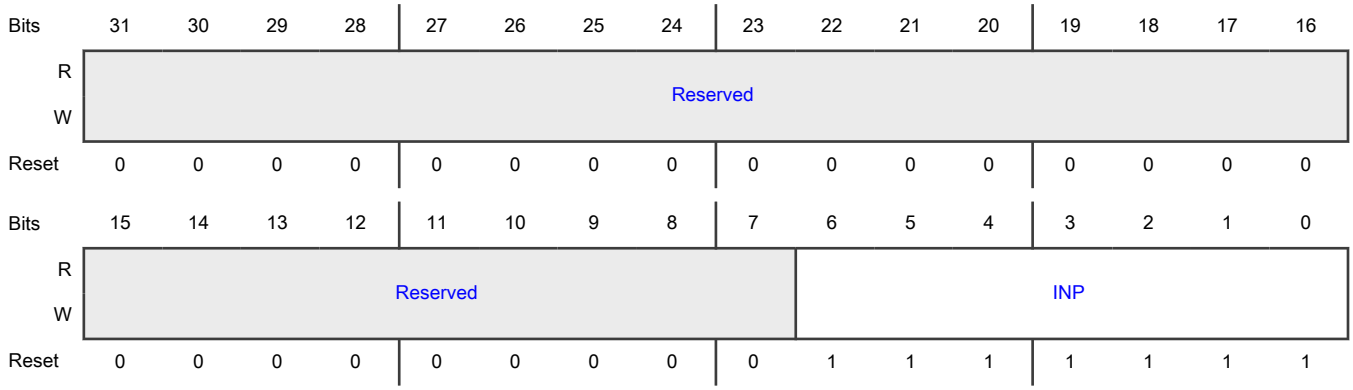
Offset

Register	Offset
CTIMER0TRIG	30h

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram



Fields

Field	Function
31-7 —	Reserved
6-0 INP	Trigger Input for CTIMER0 000_0000b - Reserved 000_0001b - CT_INP0 input is selected 000_0010b - CT_INP1 input is selected 000_0011b - CT_INP2 input is selected 000_0100b - CT_INP3 input is selected 000_0101b - CT_INP4 input is selected 000_0110b - CT_INP5 input is selected 000_0111b - CT_INP6 input is selected 000_1000b - Reserved 000_1001b - CT_INP8 input is selected 000_1010b - CT_INP9 input is selected 000_1011b - Reserved 000_1100b - Reserved 000_1101b - CT_INP12 input is selected 000_1110b - CT_INP13 input is selected 000_1111b - Reserved 001_0000b - Reserved 001_0001b - CT_INP16 input is selected 001_0010b - CT_INP17 input is selected 001_0011b - CT_INP18 input is selected

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	001_0100b - CT_INP19 input is selected 001_0101b - Reserved 001_0110b - Reserved 001_0111b - Reserved 001_1000b - Reserved 001_1001b - Reserved 001_1010b - ADC0_tcomp[0] input is selected 001_1011b - ADC0_tcomp[1] input is selected 001_1100b - ADC0_tcomp[2] input is selected 001_1101b - ADC0_tcomp[3] input is selected 001_1110b - CMP0_OUT input is selected 001_1111b - Reserved 010_0000b - Reserved 010_0001b - CTimer1_MAT1 input is selected 010_0010b - CTimer1_MAT2 input is selected 010_0011b - CTimer1_MAT3 input is selected 010_0100b - Reserved 010_0101b - Reserved 010_0110b - Reserved 010_0111b - Reserved 010_1000b - Reserved 010_1001b - Reserved 010_1010b - Reserved 010_1011b - Reserved 010_1100b - PWM0_SM0_MUX_TRIG0 input is selected 010_1101b - PWM0_SM1_MUX_TRIG0 input is selected 010_1110b - PWM0_SM2_MUX_TRIG0 input is selected 010_1111b - Reserved 011_0000b - LPI2C0 Master End of Packet input is selected 011_0001b - LPI2C0 Slave End of Packet input is selected 011_0010b - Reserved 011_0011b - Reserved 011_0100b - LPSPI0 End of Frame input is selected

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	011_0101b - LPSPI0 Received Data Word input is selected 011_0110b - Reserved 011_0111b - Reserved 011_1000b - LPUART0 Received Data Word input is selected 011_1001b - LPUART0 Transmitted Data Word input is selected 011_1010b - LPUART0 Receive Line Idle input is selected 011_1011b - LPUART1 Received Data Word input is selected 011_1100b - LPUART1 Transmitted Data Word input is selected 011_1101b - LPUART1 Receive Line Idle input is selected 011_1110b - LPUART2 Received Data Word input is selected 011_1111b - LPUART2 Transmitted Data Word input is selected 100_0000b - LPUART2 Receive Line Idle input is selected 100_0001b - TRIG_IN0 input is selected 100_0010b - TRIG_IN1 input is selected 100_0011b - TRIG_IN2 input is selected 100_0100b - TRIG_IN3 input is selected 100_0101b - TRIG_IN4 input is selected 100_0110b - TRIG_IN5 input is selected 100_0111b - TRIG_IN6 input is selected 100_1000b - TRIG_IN7 input is selected 100_1001b - TRIG_IN8 input is selected 100_1010b - TRIG_IN9 input is selected 100_1011b - TRIG_IN10 input is selected 100_1100b - TRIG_IN11 input is selected 100_1101b - LPUART3 Received Data Word input is selected 100_1110b - LPUART3 Transmitted Data Word input is selected 100_1111b - LPUART3 Receive Line Idle input is selected All other values are reserved.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

12.6.1.4 Capture Select for CTIMER1 Inputs (CTIMER1CAP0 - CTIMER1CAP3)

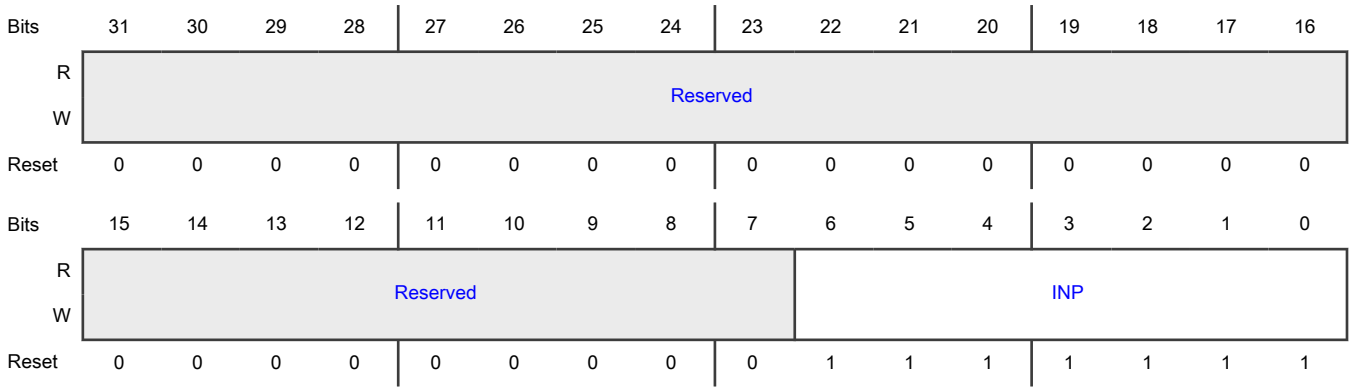
Offset

Register	Offset
CTIMER1CAP0	40h
CTIMER1CAP1	44h
CTIMER1CAP2	48h
CTIMER1CAP3	4Ch

Function

For CTIMER1, there are 4 TIMER1CAPa, with a = 0 to 3, each allowing selecting between external or internal input sources. The output of CTIMER1CAP0 register selects the source for CTIMER1 capture input 0. The output of CTIMER1CAP1 register selects the source for CTIMER1 capture input 1 and so forth up to CTIMER1CAP3 register, which selects the input for CTIMER1 capture input 3.

Diagram



Fields

Field	Function
31-7 —	Reserved
6-0 INP	Capture Input a for CTIMER1 000_0000b - Reserved 000_0001b - CT_INP0 input is selected 000_0010b - CT_INP1 input is selected 000_0011b - CT_INP2 input is selected 000_0100b - CT_INP3 input is selected 000_0101b - CT_INP4 input is selected

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	000_0110b - CT_INP5 input is selected 000_0111b - CT_INP6 input is selected 000_1000b - Reserved 000_1001b - CT_INP8 input is selected 000_1010b - CT_INP9 input is selected 000_1011b - Reserved 000_1100b - Reserved 000_1101b - CT_INP12 input is selected 000_1110b - CT_INP13 input is selected 000_1111b - Reserved 001_0000b - Reserved 001_0001b - CT_INP16 input is selected 001_0010b - CT_INP17 input is selected 001_0011b - CT_INP18 input is selected 001_0100b - CT_INP19 input is selected 001_0101b - Reserved 001_0110b - Reserved 001_0111b - Reserved 001_1000b - Reserved 001_1001b - Reserved 001_1010b - ADC0_tcomp[0] input is selected 001_1011b - ADC0_tcomp[1] input is selected 001_1100b - ADC0_tcomp[2] input is selected 001_1101b - ADC0_tcomp[3] input is selected 001_1110b - CMP0_OUT input is selected 001_1111b - Reserved 010_0000b - Reserved 010_0001b - CTimer0_MAT1 input is selected 010_0010b - CTimer0_MAT2 input is selected 010_0011b - CTimer0_MAT3 input is selected 010_0100b - Reserved 010_0101b - Reserved 010_0110b - Reserved

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	010_0111b - Reserved 010_1000b - Reserved 010_1001b - Reserved 010_1010b - Reserved 010_1011b - Reserved 010_1100b - PWM0_SM0_MUX_TRIG0 input is selected 010_1101b - PWM0_SM1_MUX_TRIG0 input is selected 010_1110b - PWM0_SM2_MUX_TRIG0 input is selected 010_1111b - Reserved 011_0000b - LPI2C0 Master End of Packet input is selected 011_0001b - LPI2C0 Slave End of Packet input is selected 011_0010b - Reserved 011_0011b - Reserved 011_0100b - LPSPI0 End of Frame input is selected 011_0101b - LPSPI0 Received Data Word input is selected 011_0110b - Reserved 011_0111b - Reserved 011_1000b - LPUART0 Received Data Word input is selected 011_1001b - LPUART0 Transmitted Data Word input is selected 011_1010b - LPUART0 Receive Line Idle input is selected 011_1011b - LPUART1 Received Data Word input is selected 011_1100b - LPUART1 Transmitted Data Word input is selected 011_1101b - LPUART1 Receive Line Idle input is selected 011_1110b - LPUART2 Received Data Word input is selected 011_1111b - LPUART2 Transmitted Data Word input is selected 100_0000b - LPUART2 Receive Line Idle input is selected 100_0001b - TRIG_IN0 input is selected 100_0010b - TRIG_IN1 input is selected 100_0011b - TRIG_IN2 input is selected 100_0100b - TRIG_IN3 input is selected 100_0101b - TRIG_IN4 input is selected 100_0110b - TRIG_IN5 input is selected 100_0111b - TRIG_IN6 input is selected

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	100_1000b - TRIG_IN7 input is selected 100_1001b - TRIG_IN8 input is selected 100_1010b - TRIG_IN9 input is selected 100_1011b - TRIG_IN10 input is selected 100_1100b - TRIG_IN11 input is selected 100_1101b - LPUART3 Received Data Word input is selected 100_1110b - LPUART3 Transmitted Data Word input is selected 100_1111b - LPUART3 Receive Line Idle input is selected All other values are reserved.

12.6.1.5 Trigger for CTIMER1 (CTIMER1TRIG)

Offset

Register	Offset
CTIMER1TRIG	50h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								INP							
Reset	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1

Fields

Field	Function
31-7 —	Reserved
6-0 INP	Trigger Input for CTIMER1

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	000_0000b - Reserved 000_0001b - CT_INP0 input is selected 000_0010b - CT_INP1 input is selected 000_0011b - CT_INP2 input is selected 000_0100b - CT_INP3 input is selected 000_0101b - CT_INP4 input is selected 000_0110b - CT_INP5 input is selected 000_0111b - CT_INP6 input is selected 000_1000b - Reserved 000_1001b - CT_INP8 input is selected 000_1010b - CT_INP9 input is selected 000_1011b - Reserved 000_1100b - Reserved 000_1101b - CT_INP12 input is selected 000_1110b - CT_INP13 input is selected 000_1111b - Reserved 001_0000b - Reserved 001_0001b - CT_INP16 input is selected 001_0010b - CT_INP17 input is selected 001_0011b - CT_INP18 input is selected 001_0100b - CT_INP19 input is selected 001_0101b - Reserved 001_0110b - Reserved 001_0111b - Reserved 001_1000b - Reserved 001_1001b - Reserved 001_1010b - ADC0_tcomp[0] input is selected 001_1011b - ADC0_tcomp[1] input is selected 001_1100b - ADC0_tcomp[2] input is selected 001_1101b - ADC0_tcomp[3] input is selected 001_1110b - CMP0_OUT input is selected 001_1111b - Reserved 010_0000b - Reserved

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	010_0001b - CTimer0_MAT1 input is selected 010_0010b - CTimer0_MAT2 input is selected 010_0011b - CTimer0_MAT3 input is selected 010_0100b - Reserved 010_0101b - Reserved 010_0110b - Reserved 010_0111b - Reserved 010_1000b - Reserved 010_1001b - Reserved 010_1010b - Reserved 010_1011b - Reserved 010_1100b - PWM0_SM0_MUX_TRIG0 input is selected 010_1101b - PWM0_SM1_MUX_TRIG0 input is selected 010_1110b - PWM0_SM2_MUX_TRIG0 input is selected 010_1111b - Reserved 011_0000b - LPI2C0 Master End of Packet input is selected 011_0001b - LPI2C0 Slave End of Packet input is selected 011_0010b - Reserved 011_0011b - Reserved 011_0100b - LPSPI0 End of Frame input is selected 011_0101b - LPSPI0 Received Data Word input is selected 011_0110b - Reserved 011_0111b - Reserved 011_1000b - LPUART0 Received Data Word input is selected 011_1001b - LPUART0 Transmitted Data Word input is selected 011_1010b - LPUART0 Receive Line Idle input is selected 011_1011b - LPUART1 Received Data Word input is selected 011_1100b - LPUART1 Transmitted Data Word input is selected 011_1101b - LPUART1 Receive Line Idle input is selected 011_1110b - LPUART2 Received Data Word input is selected 011_1111b - LPUART2 Transmitted Data Word input is selected 100_0000b - LPUART2 Receive Line Idle input is selected 100_0001b - TRIG_IN0 input is selected

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	100_0010b - TRIG_IN1 input is selected 100_0011b - TRIG_IN2 input is selected 100_0100b - TRIG_IN3 input is selected 100_0101b - TRIG_IN4 input is selected 100_0110b - TRIG_IN5 input is selected 100_0111b - TRIG_IN6 input is selected 100_1000b - TRIG_IN7 input is selected 100_1001b - TRIG_IN8 input is selected 100_1010b - TRIG_IN9 input is selected 100_1011b - TRIG_IN10 input is selected 100_1100b - TRIG_IN11 input is selected 100_1101b - LPUART3 Received Data Word input is selected 100_1110b - LPUART3 Transmitted Data Word input is selected 100_1111b - LPUART3 Receive Line Idle input is selected All other values are reserved.

12.6.1.6 FREQME Reference Clock Selection (FREQMEAS_REF)

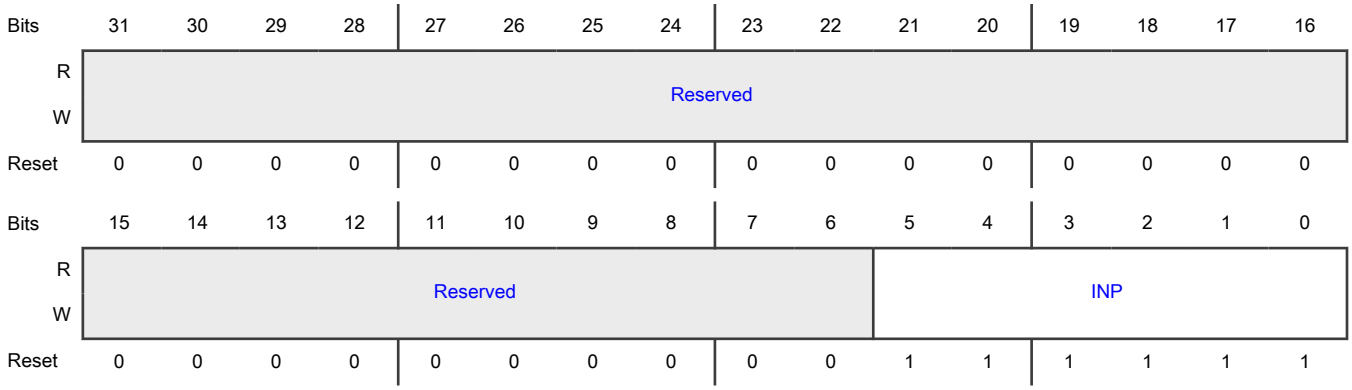
Offset

Register	Offset
FREQMEAS_REF	180h

Function

Selects a clock as the reference clock of the frequency measure function.

Diagram



All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Fields

Field	Function
31-6 —	Reserved
5-0 INP	Clock source (binary value) for frequency measure function reference clock. 00_0000b - Reserved 00_0001b - clk_in input is selected 00_0010b - FRO_OSC_12M input is selected 00_0011b - fro_hf_div input is selected 00_0100b - Osc32k[1] input is selected 00_0101b - clk_16k[1] input is selected 00_0110b - SLOW_CLK input is selected 00_0111b - FREQME_CLK_IN0 input is selected 00_1000b - FREQME_CLK_IN1 input is selected 00_1001b - Reserved 00_1010b - Reserved 00_1011b - PWM0_SM0_MUX_TRIG0 input is selected 00_1100b - PWM0_SM0_MUX_TRIG1 input is selected 00_1101b - PWM0_SM1_MUX_TRIG0 input is selected 00_1110b - PWM0_SM1_MUX_TRIG1 input is selected 00_1111b - PWM0_SM2_MUX_TRIG0 input is selected 01_0000b - PWM0_SM2_MUX_TRIG1 input is selected All other values are reserved.

12.6.1.7 FREQME Target Clock Selection (FREQMEAS_TAR)

Offset

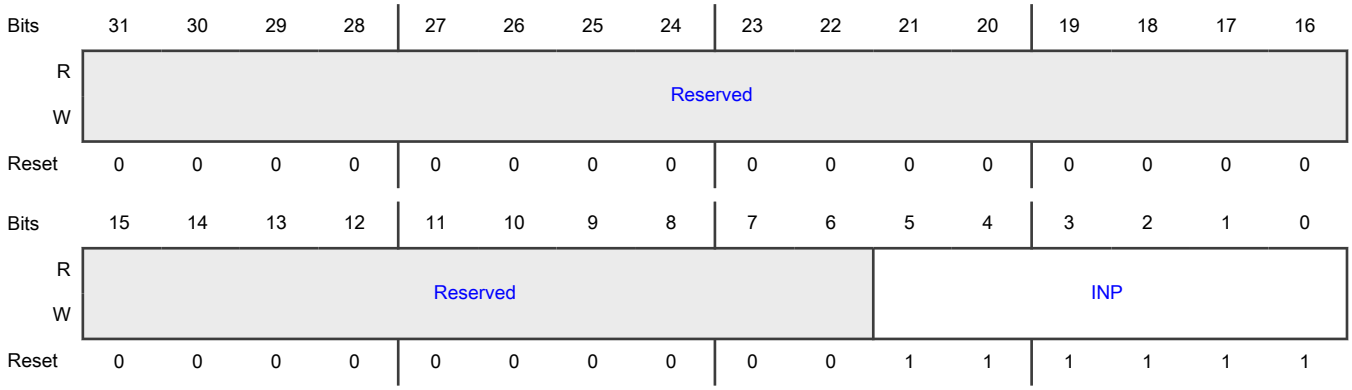
Register	Offset
FREQMEAS_TAR	184h

Function
Selects a clock as the target clock of the frequency measure function.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram



Fields

Field	Function
31-6 —	Reserved
5-0 INP	Clock source (binary value) for frequency measure function target clock. 00_0000b - Reserved 00_0001b - clk_in input is selected 00_0010b - FRO_OSC_12M input is selected 00_0011b - fro_hf_div input is selected 00_0100b - Osc32k[1] input is selected 00_0101b - clk_16k[1] input is selected 00_0110b - SLOW_CLK input is selected 00_0111b - FREQME_CLK_IN0 input is selected 00_1000b - FREQME_CLK_IN1 input is selected 00_1001b - Reserved 00_1010b - Reserved 00_1011b - PWM0_SM0_MUX_TRIG0 input is selected 00_1100b - PWM0_SM0_MUX_TRIG1 input is selected 00_1101b - PWM0_SM1_MUX_TRIG0 input is selected 00_1110b - PWM0_SM1_MUX_TRIG1 input is selected 00_1111b - PWM0_SM2_MUX_TRIG0 input is selected 01_0000b - PWM0_SM2_MUX_TRIG1 input is selected All other values are reserved.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

12.6.1.8 LPCMP0 Input Connections (CMP0_TRIG)

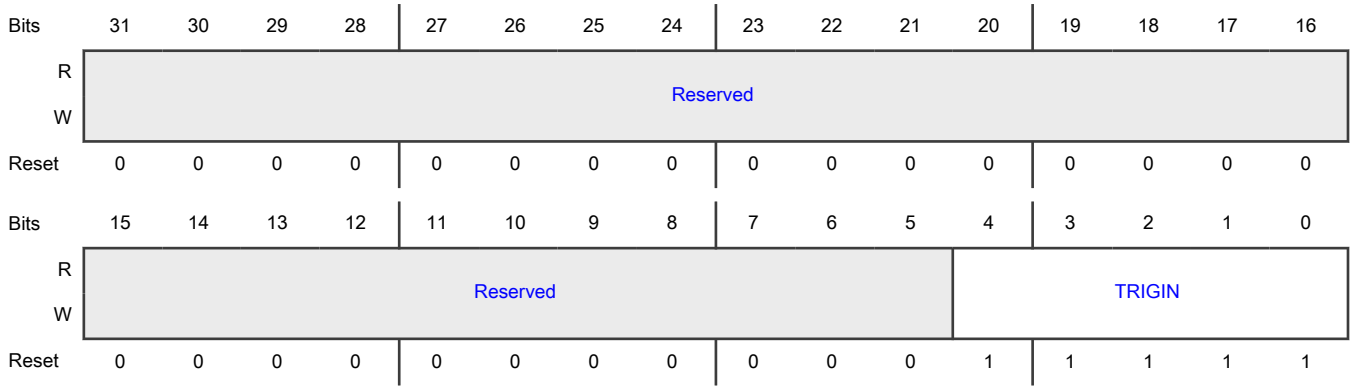
Offset

Register	Offset
CMP0_TRIG	260h

Function

Selects the LPCMP0 trigger inputs.

Diagram



Fields

Field	Function
31-5 —	Reserved
4-0 TRIGIN	LPCMP0 Input Trigger 0_0000b - Reserved 0_0001b - Reserved 0_0010b - Reserved 0_0011b - Reserved 0_0100b - Reserved 0_0101b - Reserved 0_0110b - Reserved 0_0111b - Reserved 0_1000b - CTimer0_MAT0 input is selected 0_1001b - CTimer0_MAT2 input is selected 0_1010b - CTimer1_MAT0 input is selected

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	0_1011b - CTimer1_MAT2 input is selected 0_1100b - Reserved 0_1101b - Reserved 0_1110b - LPTMR0 input is selected 0_1111b - Reserved 1_0000b - Reserved 1_0001b - PWM0_SM0_MUX_TRIG0 input is selected 1_0010b - PWM0_SM0_MUX_TRIG1 input is selected 1_0011b - PWM0_SM1_MUX_TRIG0 input is selected 1_0100b - PWM0_SM1_MUX_TRIG1 input is selected 1_0101b - PWM0_SM2_MUX_TRIG0 input is selected 1_0110b - PWM0_SM2_MUX_TRIG1 input is selected 1_0111b - Reserved 1_1000b - Reserved 1_1001b - GPIO0 Pin Event Trig 0 input is selected 1_1010b - GPIO1 Pin Event Trig 0 input is selected 1_1011b - GPIO2 Pin Event Trig 0 input is selected 1_1100b - GPIO3 Pin Event Trig 0 input is selected 1_1101b - Reserved 1_1110b - WUU input is selected All other values are reserved.

12.6.1.9 ADC0 Trigger Input Connections (ADC0_TRIG0 - ADC0_TRIG3)

Offset

Register	Offset
ADC0_TRIG0	280h
ADC0_TRIG1	284h
ADC0_TRIG2	288h
ADC0_TRIG3	28Ch

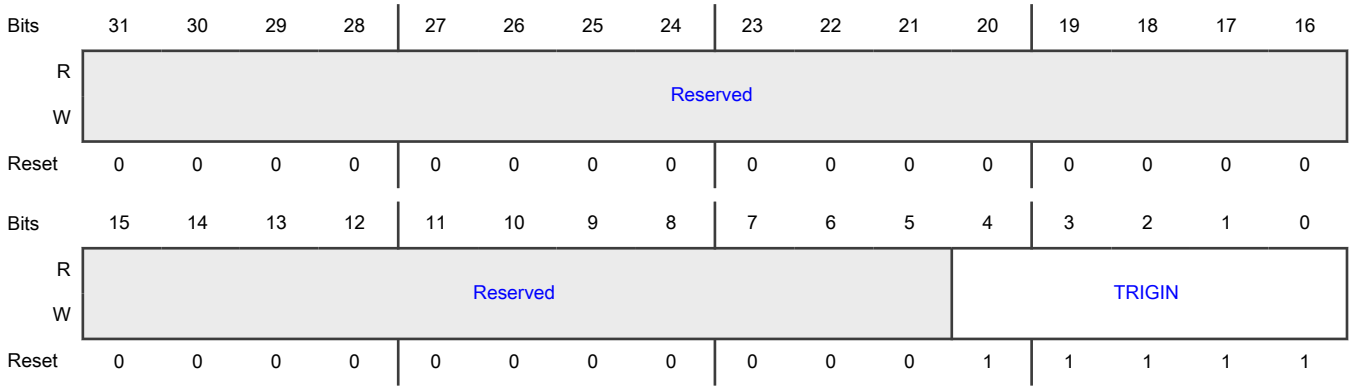
Function

Selects the ADC0 trigger a inputs.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram



Fields

Field	Function
31-5 —	Reserved
4-0 TRIGIN	ADC0 Trigger a Inputs 0_0000b - Reserved 0_0001b - ARM_TXEV input is selected 0_0010b - Reserved 0_0011b - Reserved 0_0100b - Reserved 0_0101b - Reserved 0_0110b - CMP0_OUT input is selected 0_0111b - Reserved 0_1000b - Reserved 0_1001b - CTimer0_MAT0 input is selected 0_1010b - CTimer0_MAT1 input is selected 0_1011b - CTimer1_MAT0 input is selected 0_1100b - CTimer1_MAT1 input is selected 0_1101b - Reserved 0_1110b - Reserved 0_1111b - LPTMR0 input is selected 1_0000b - Reserved 1_0001b - Reserved 1_0010b - PWM0_SM0_MUX_TRIG0 input is selected 1_0011b - PWM0_SM0_MUX_TRIG1 input is selected

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	1_0100b - PWM0_SM1_MUX_TRIG0 input is selected 1_0101b - PWM0_SM1_MUX_TRIG1 input is selected 1_0110b - PWM0_SM2_MUX_TRIG0 input is selected 1_0111b - PWM0_SM2_MUX_TRIG1 input is selected 1_1000b - Reserved 1_1001b - Reserved 1_1010b - GPIO0 Pin Event Trig 0 input is selected 1_1011b - GPIO1 Pin Event Trig 0 input is selected 1_1100b - GPIO2 Pin Event Trig 0 input is selected 1_1101b - GPIO3 Pin Event Trig 0 input is selected 1_1110b - Reserved 1_1111b - WUU input is selected All other values are reserved.

12.6.1.10 eFlexPWM0 SM0 EXTA Input Trigger Connections (FlexPWM0_SM0_EXTA)

Offset

Register	Offset
FlexPWM0_SM0_EXTA	3A0h

Function

This register selects the PWM0 trigger inputs.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved										TRIGIN					
Reset	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Fields

Field	Function
31-6 —	Reserved
5-0 TRIGIN	Trigger Input Connections for eFlexPWM0 SM0 EXTA 00_0000b - Reserved 00_0001b - ARM_TXEV input is selected 00_0010b - Reserved 00_0011b - Reserved 00_0100b - Reserved 00_0101b - Reserved 00_0110b - CMP0_OUT input is selected 00_0111b - Reserved 00_1000b - Reserved 00_1001b - CTimer0_MAT2 input is selected 00_1010b - CTimer0_MAT3 input is selected 00_1011b - CTimer1_MAT2 input is selected 00_1100b - CTimer1_MAT3 input is selected 00_1101b - Reserved 00_1110b - Reserved 00_1111b - Reserved 01_0000b - Reserved 01_0001b - Reserved 01_0010b - Reserved 01_0011b - Reserved 01_0100b - TRIG_IN0 input is selected 01_0101b - TRIG_IN1 input is selected 01_0110b - TRIG_IN2 input is selected 01_0111b - TRIG_IN3 input is selected 01_1000b - TRIG_IN4 input is selected 01_1001b - TRIG_IN5 input is selected 01_1010b - TRIG_IN6 input is selected 01_1011b - TRIG_IN7 input is selected 01_1100b - TRIG_IN8 input is selected 01_1101b - TRIG_IN9 input is selected

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	01_1110b - TRIG_IN10 input is selected 01_1111b - TRIG_IN11 input is selected 10_0000b - GPIO0 Pin Event Trig 0 input is selected 10_0001b - GPIO1 Pin Event Trig 0 input is selected 10_0010b - GPIO2 Pin Event Trig 0 input is selected 10_0011b - GPIO3 Pin Event Trig 0 input is selected 10_0100b - Reserved 10_0101b - ECC Error Interrupt input is selected 10_0110b - WWDT0/1 Interrupt input is selected 10_0111b - Reserved 10_1000b - Reserved 10_1001b - Reserved 10_1010b - Reserved 10_1011b - Reserved 10_1100b - Reserved 10_1101b - Reserved 10_1110b - Reserved 10_1111b - Reserved 11_0000b - Reserved 11_0001b - Reserved 11_0010b - Reserved 11_0011b - Reserved 11_0100b - Reserved 11_0101b - Reserved 11_0110b - Reserved 11_0111b - Reserved 11_1000b - Reserved 11_1001b - Reserved 11_1010b - Reserved 11_1011b - Reserved 11_1100b - Reserved 11_1101b - Reserved All other values are reserved.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

12.6.1.11 eFlexPWM0 SM0 EXT_SYNC Input Trigger Connections (FlexPWM0_SM0_EXTSYNC)

Offset

Register	Offset
FlexPWM0_SM0_EXTSYNC	3A4h

Function
Selects the eFlexPWM0 SM0 EXT_SYNC trigger inputs.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved										TRIGIN					
Reset	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1

Fields

Field	Function
31-6 —	Reserved
5-0 TRIGIN	Input Trigger Connections for eFlexPWM0 SM0 EXT_SYNC 00_0000b - Reserved 00_0001b - ARM_TXEV input is selected 00_0010b - Reserved 00_0011b - Reserved 00_0100b - Reserved 00_0101b - Reserved 00_0110b - CMP0_OUT input is selected 00_0111b - Reserved 00_1000b - Reserved 00_1001b - CTimer0_MAT2 input is selected

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	00_1010b - CTimer0_MAT3 input is selected 00_1011b - CTimer1_MAT2 input is selected 00_1100b - CTimer1_MAT3 input is selected 00_1101b - Reserved 00_1110b - Reserved 00_1111b - Reserved 01_0000b - Reserved 01_0001b - Reserved 01_0010b - Reserved 01_0011b - Reserved 01_0100b - TRIG_IN0 input is selected 01_0101b - TRIG_IN1 input is selected 01_0110b - TRIG_IN2 input is selected 01_0111b - TRIG_IN3 input is selected 01_1000b - TRIG_IN4 input is selected 01_1001b - TRIG_IN5 input is selected 01_1010b - TRIG_IN6 input is selected 01_1011b - TRIG_IN7 input is selected 01_1100b - TRIG_IN8 input is selected 01_1101b - TRIG_IN9 input is selected 01_1110b - TRIG_IN10 input is selected 01_1111b - TRIG_IN11 input is selected 10_0000b - GPIO0 Pin Event Trig 0 input is selected 10_0001b - GPIO1 Pin Event Trig 0 input is selected 10_0010b - GPIO2 Pin Event Trig 0 input is selected 10_0011b - GPIO3 Pin Event Trig 0 input is selected 10_0100b - Reserved 10_0101b - ECC Error Interrupt input is selected 10_0110b - WWDT0/1 Interrupt input is selected 10_0111b - Reserved 10_1000b - Reserved 10_1001b - Reserved 10_1010b - Reserved

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	10_1011b - Reserved 10_1100b - Reserved 10_1101b - Reserved 10_1110b - Reserved 10_1111b - Reserved 11_0000b - Reserved 11_0001b - Reserved 11_0010b - Reserved 11_0011b - Reserved 11_0100b - Reserved 11_0101b - Reserved 11_0110b - Reserved 11_0111b - Reserved 11_1000b - Reserved 11_1001b - Reserved 11_1010b - Reserved 11_1011b - Reserved 11_1100b - Reserved 11_1101b - Reserved All other values are reserved.

12.6.1.12 eFlexPWM0 SM1 EXTA Input Trigger Connections (FlexPWM0_SM1_EXTA)

Offset

Register	Offset
FlexPWM0_SM1_EXTA	3A8h

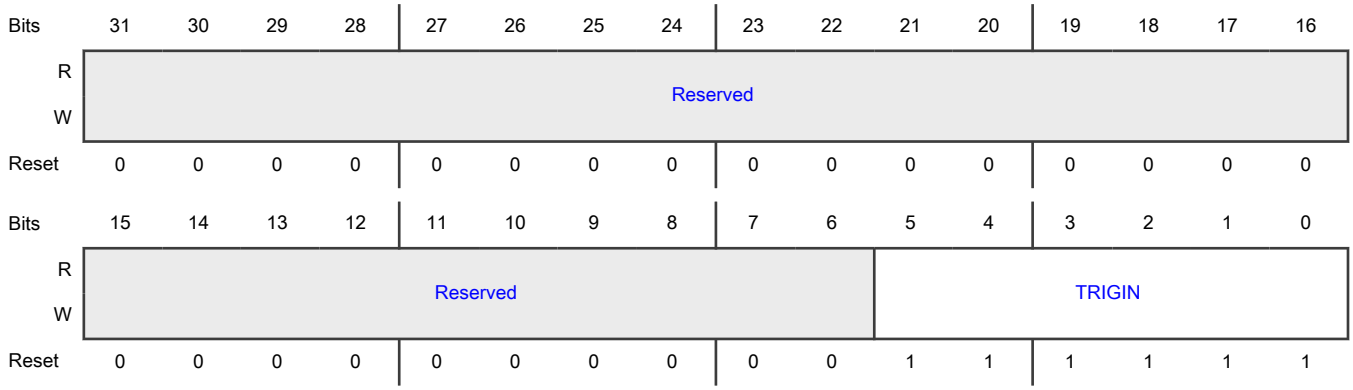
Function

Selects the eFlexPWM0 SM1 EXTA trigger inputs.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram



Fields

Field	Function
31-6 —	Reserved
5-0 TRIGIN	Input Trigger Connections for eFlexPWM0 SM1 EXTA 00_0000b - Reserved 00_0001b - ARM_TXEV input is selected 00_0010b - Reserved 00_0011b - Reserved 00_0100b - Reserved 00_0101b - Reserved 00_0110b - CMP0_OUT input is selected 00_0111b - Reserved 00_1000b - Reserved 00_1001b - CTimer0_MAT2 input is selected 00_1010b - CTimer0_MAT3 input is selected 00_1011b - CTimer1_MAT2 input is selected 00_1100b - CTimer1_MAT3 input is selected 00_1101b - Reserved 00_1110b - Reserved 00_1111b - Reserved 01_0000b - Reserved 01_0001b - Reserved 01_0010b - Reserved 01_0011b - Reserved

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	01_0100b - TRIG_IN0 input is selected 01_0101b - TRIG_IN1 input is selected 01_0110b - TRIG_IN2 input is selected 01_0111b - TRIG_IN3 input is selected 01_1000b - TRIG_IN4 input is selected 01_1001b - TRIG_IN5 input is selected 01_1010b - TRIG_IN6 input is selected 01_1011b - TRIG_IN7 input is selected 01_1100b - TRIG_IN8 input is selected 01_1101b - TRIG_IN9 input is selected 01_1110b - TRIG_IN10 input is selected 01_1111b - TRIG_IN11 input is selected 10_0000b - GPIO0 Pin Event Trig 0 input is selected 10_0001b - GPIO1 Pin Event Trig 0 input is selected 10_0010b - GPIO2 Pin Event Trig 0 input is selected 10_0011b - GPIO3 Pin Event Trig 0 input is selected 10_0100b - Reserved 10_0101b - ECC Error Interrupt input is selected 10_0110b - WWDT0/1 Interrupt input is selected 10_0111b - Reserved 10_1000b - Reserved 10_1001b - Reserved 10_1010b - Reserved 10_1011b - Reserved 10_1100b - Reserved 10_1101b - Reserved 10_1110b - Reserved 10_1111b - Reserved 11_0000b - Reserved 11_0001b - Reserved 11_0010b - Reserved 11_0011b - Reserved 11_0100b - Reserved

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	11_0101b - Reserved 11_0110b - Reserved 11_0111b - Reserved 11_1000b - Reserved 11_1001b - Reserved 11_1010b - Reserved 11_1011b - Reserved 11_1100b - Reserved 11_1101b - Reserved All other values are reserved.

12.6.1.13 eFlexPWM0 SM1 EXT_SYNC Input Trigger Connections (FlexPWM0_SM1_EXTSYNC)

Offset

Register	Offset
FlexPWM0_SM1_EXTSYNC	3ACh

Function

Selects the eFlexPWM0 SM01 EXT_SYNC trigger inputs.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved										TRIGIN					
Reset	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Fields

Field	Function
31-6 —	Reserved
5-0 TRIGIN	Input Trigger Connections for eFlexPWM0 SM1 EXT_SYNC 00_0000b - Reserved 00_0001b - ARM_TXEV input is selected 00_0010b - Reserved 00_0011b - Reserved 00_0100b - Reserved 00_0101b - Reserved 00_0110b - CMP0_OUT input is selected 00_0111b - Reserved 00_1000b - Reserved 00_1001b - CTimer0_MAT2 input is selected 00_1010b - CTimer0_MAT3 input is selected 00_1011b - CTimer1_MAT2 input is selected 00_1100b - CTimer1_MAT3 input is selected 00_1101b - Reserved 00_1110b - Reserved 00_1111b - Reserved 01_0000b - Reserved 01_0001b - Reserved 01_0010b - Reserved 01_0011b - Reserved 01_0100b - TRIG_IN0 input is selected 01_0101b - TRIG_IN1 input is selected 01_0110b - TRIG_IN2 input is selected 01_0111b - TRIG_IN3 input is selected 01_1000b - TRIG_IN4 input is selected 01_1001b - TRIG_IN5 input is selected 01_1010b - TRIG_IN6 input is selected 01_1011b - TRIG_IN7 input is selected 01_1100b - TRIG_IN8 input is selected 01_1101b - TRIG_IN9 input is selected

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	01_1110b - TRIG_IN10 input is selected 01_1111b - TRIG_IN11 input is selected 10_0000b - GPIO0 Pin Event Trig 0 input is selected 10_0001b - GPIO1 Pin Event Trig 0 input is selected 10_0010b - GPIO2 Pin Event Trig 0 input is selected 10_0011b - GPIO3 Pin Event Trig 0 input is selected 10_0100b - Reserved 10_0101b - ECC Error Interrupt input is selected 10_0110b - WWDT0/1 Interrupt input is selected 10_0111b - Reserved 10_1000b - Reserved 10_1001b - Reserved 10_1010b - Reserved 10_1011b - Reserved 10_1100b - Reserved 10_1101b - Reserved 10_1110b - Reserved 10_1111b - Reserved 11_0000b - Reserved 11_0001b - Reserved 11_0010b - Reserved 11_0011b - Reserved 11_0100b - Reserved 11_0101b - Reserved 11_0110b - Reserved 11_0111b - Reserved 11_1000b - Reserved 11_1001b - Reserved 11_1010b - Reserved 11_1011b - Reserved 11_1100b - Reserved 11_1101b - Reserved All other values are reserved.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

12.6.1.14 eFlexPWM0 SM2 EXTA Input Trigger Connections (FlexPWM0_SM2_EXTA)

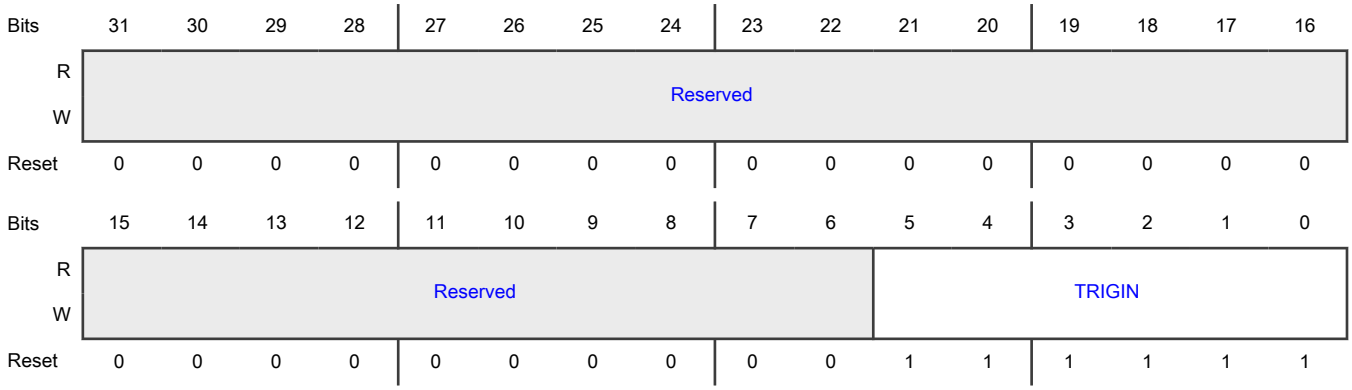
Offset

Register	Offset
FlexPWM0_SM2_EXTA	3B0h

Function

Selects the eFlexPWM0 SM2 EXTA trigger inputs.

Diagram



Fields

Field	Function
31-6 —	Reserved
5-0 TRIGIN	Input Trigger Connections for eFlexPWM0 SM2 EXTA 00_0000b - Reserved 00_0001b - ARM_TXEV input is selected 00_0010b - Reserved 00_0011b - Reserved 00_0100b - Reserved 00_0101b - Reserved 00_0110b - CMP0_OUT input is selected 00_0111b - Reserved 00_1000b - Reserved 00_1001b - CTimer0_MAT2 input is selected 00_1010b - CTimer0_MAT3 input is selected

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	00_1011b - CTimer1_MAT2 input is selected 00_1100b - CTimer1_MAT3 input is selected 00_1101b - Reserved 00_1110b - Reserved 00_1111b - Reserved 01_0000b - Reserved 01_0001b - Reserved 01_0010b - Reserved 01_0011b - Reserved 01_0100b - TRIG_IN0 input is selected 01_0101b - TRIG_IN1 input is selected 01_0110b - TRIG_IN2 input is selected 01_0111b - TRIG_IN3 input is selected 01_1000b - TRIG_IN4 input is selected 01_1001b - TRIG_IN5 input is selected 01_1010b - TRIG_IN6 input is selected 01_1011b - TRIG_IN7 input is selected 01_1100b - TRIG_IN8 input is selected 01_1101b - TRIG_IN9 input is selected 01_1110b - TRIG_IN10 input is selected 01_1111b - TRIG_IN11 input is selected 10_0000b - GPIO0 Pin Event Trig 0 input is selected 10_0001b - GPIO1 Pin Event Trig 0 input is selected 10_0010b - GPIO2 Pin Event Trig 0 input is selected 10_0011b - GPIO3 Pin Event Trig 0 input is selected 10_0100b - Reserved 10_0101b - ECC Error Interrupt input is selected 10_0110b - WWDT0/1 Interrupt input is selected 10_0111b - Reserved 10_1000b - Reserved 10_1001b - Reserved 10_1010b - Reserved 10_1011b - Reserved

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	10_1100b - Reserved 10_1101b - Reserved 10_1110b - Reserved 10_1111b - Reserved 11_0000b - Reserved 11_0001b - Reserved 11_0010b - Reserved 11_0011b - Reserved 11_0100b - Reserved 11_0101b - Reserved 11_0110b - Reserved 11_0111b - Reserved 11_1000b - Reserved 11_1001b - Reserved 11_1010b - Reserved 11_1011b - Reserved 11_1100b - Reserved 11_1101b - Reserved All other values are reserved.

12.6.1.15 eFlexPWM0 SM2 EXT_SYNC Input Trigger Connections (FlexPWM0_SM2_EXTSYNC)

Offset

Register	Offset
FlexPWM0_SM2_EXTSYNC	3B4h

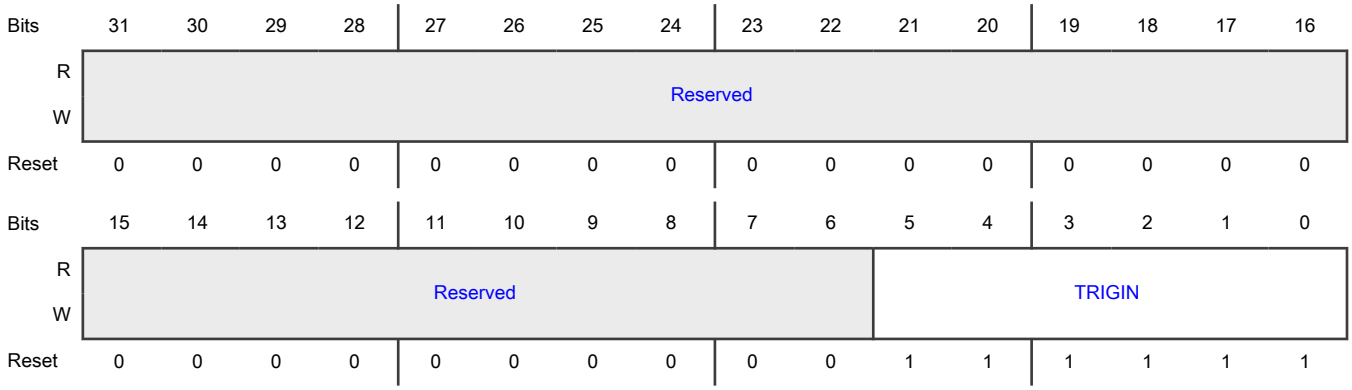
Function

Selects the eFlexPWM2 SM0 EXT_SYNC trigger inputs.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram



Fields

Field	Function
31-6 —	Reserved
5-0 TRIGIN	Input Trigger Connections for eFlexPWM0 SM2 EXT_SYNC 00_0000b - Reserved 00_0001b - ARM_TXEV input is selected 00_0010b - Reserved 00_0011b - Reserved 00_0100b - Reserved 00_0101b - Reserved 00_0110b - CMP0_OUT input is selected 00_0111b - Reserved 00_1000b - Reserved 00_1001b - CTimer0_MAT2 input is selected 00_1010b - CTimer0_MAT3 input is selected 00_1011b - CTimer1_MAT2 input is selected 00_1100b - CTimer1_MAT3 input is selected 00_1101b - Reserved 00_1110b - Reserved 00_1111b - Reserved 01_0000b - Reserved 01_0001b - Reserved 01_0010b - Reserved 01_0011b - Reserved

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	01_0100b - TRIG_IN0 input is selected 01_0101b - TRIG_IN1 input is selected 01_0110b - TRIG_IN2 input is selected 01_0111b - TRIG_IN3 input is selected 01_1000b - TRIG_IN4 input is selected 01_1001b - TRIG_IN5 input is selected 01_1010b - TRIG_IN6 input is selected 01_1011b - TRIG_IN7 input is selected 01_1100b - TRIG_IN8 input is selected 01_1101b - TRIG_IN9 input is selected 01_1110b - TRIG_IN10 input is selected 01_1111b - TRIG_IN11 input is selected 10_0000b - GPIO0 Pin Event Trig 0 input is selected 10_0001b - GPIO1 Pin Event Trig 0 input is selected 10_0010b - GPIO2 Pin Event Trig 0 input is selected 10_0011b - GPIO3 Pin Event Trig 0 input is selected 10_0100b - Reserved 10_0101b - ECC Error Interrupt input is selected 10_0110b - WWDT0/1 Interrupt input is selected 10_0111b - Reserved 10_1000b - Reserved 10_1001b - Reserved 10_1010b - Reserved 10_1011b - Reserved 10_1100b - Reserved 10_1101b - Reserved 10_1110b - Reserved 10_1111b - Reserved 11_0000b - Reserved 11_0001b - Reserved 11_0010b - Reserved 11_0011b - Reserved 11_0100b - Reserved

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	11_0101b - Reserved 11_0110b - Reserved 11_0111b - Reserved 11_1000b - Reserved 11_1001b - Reserved 11_1010b - Reserved 11_1011b - Reserved 11_1100b - Reserved 11_1101b - Reserved All other values are reserved.

12.6.1.16 eFlexPWM0 FAULT0 Input Trigger Connections (FlexPWM0_FAULT0)

Offset

Register	Offset
FlexPWM0_FAULT0	3C0h

Function

Selects the eFlexPWM0 FAULT0 trigger inputs.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved										TRIGIN					
Reset	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1

Fields

Field	Function
31-6	Reserved

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
—	
5-0 TRIGIN	Input Trigger Connections for eFlexPWM0 FAULT0 00_0000b - Reserved 00_0001b - ARM_TXEV input is selected 00_0010b - Reserved 00_0011b - Reserved 00_0100b - Reserved 00_0101b - Reserved 00_0110b - CMP0_OUT input is selected 00_0111b - Reserved 00_1000b - Reserved 00_1001b - CTimer0_MAT2 input is selected 00_1010b - CTimer0_MAT3 input is selected 00_1011b - CTimer1_MAT2 input is selected 00_1100b - CTimer1_MAT3 input is selected 00_1101b - Reserved 00_1110b - Reserved 00_1111b - Reserved 01_0000b - Reserved 01_0001b - Reserved 01_0010b - Reserved 01_0011b - Reserved 01_0100b - TRIG_IN0 input is selected 01_0101b - TRIG_IN1 input is selected 01_0110b - TRIG_IN2 input is selected 01_0111b - TRIG_IN3 input is selected 01_1000b - TRIG_IN4 input is selected 01_1001b - TRIG_IN5 input is selected 01_1010b - TRIG_IN6 input is selected 01_1011b - TRIG_IN7 input is selected 01_1100b - TRIG_IN8 input is selected 01_1101b - TRIG_IN9 input is selected 01_1110b - TRIG_IN10 input is selected

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	01_1111b - TRIG_IN11 input is selected 10_0000b - GPIO0 Pin Event Trig 0 input is selected 10_0001b - GPIO1 Pin Event Trig 0 input is selected 10_0010b - GPIO2 Pin Event Trig 0 input is selected 10_0011b - GPIO3 Pin Event Trig 0 input is selected 10_0100b - Reserved 10_0101b - ECC Error Interrupt input is selected 10_0110b - WWDT0/1 Interrupt input is selected 10_0111b - Reserved 10_1000b - Reserved 10_1001b - Reserved 10_1010b - Reserved 10_1011b - Reserved 10_1100b - Reserved 10_1101b - Reserved 10_1110b - Reserved 10_1111b - Reserved 11_0000b - Reserved 11_0001b - Reserved 11_0010b - Reserved 11_0011b - Reserved 11_0100b - Reserved 11_0101b - Reserved 11_0110b - Reserved 11_0111b - Reserved 11_1000b - Reserved 11_1001b - Reserved 11_1010b - Reserved 11_1011b - Reserved 11_1100b - Reserved 11_1101b - Reserved All other values are reserved.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

12.6.1.17 eFlexPWM0 FAULT1 Input Trigger Connections (FlexPWM0_FAULT1)

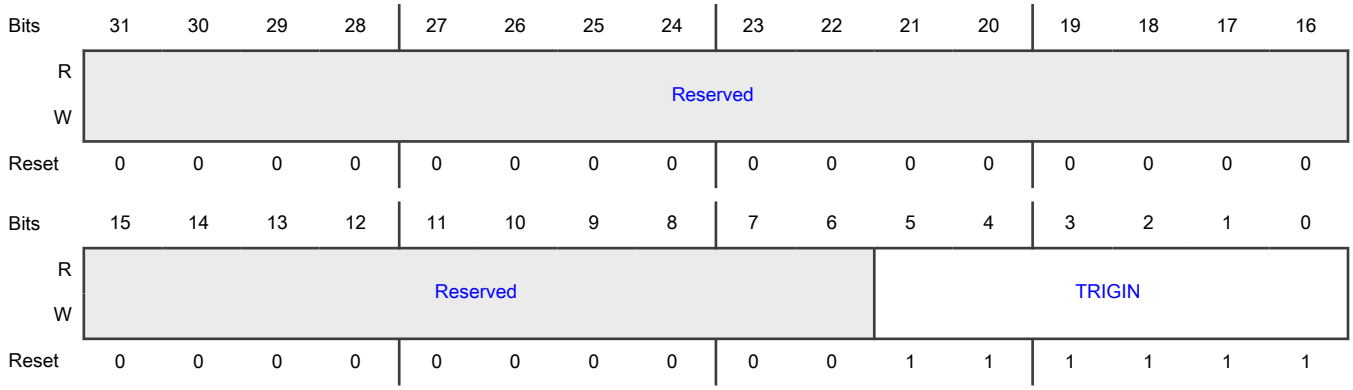
Offset

Register	Offset
FlexPWM0_FAULT1	3C4h

Function

Selects the eFlexPWM0 FAULT1 trigger inputs.

Diagram



Fields

Field	Function
31-6 —	Reserved
5-0 TRIGIN	Input Trigger Connections for eFlexPWM0 FAULT1 00_0000b - Reserved 00_0001b - ARM_TXEV input is selected 00_0010b - Reserved 00_0011b - Reserved 00_0100b - Reserved 00_0101b - Reserved 00_0110b - CMP0_OUT input is selected 00_0111b - Reserved 00_1000b - Reserved 00_1001b - CTimer0_MAT2 input is selected 00_1010b - CTimer0_MAT3 input is selected

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	00_1011b - CTimer1_MAT2 input is selected 00_1100b - CTimer1_MAT3 input is selected 00_1101b - Reserved 00_1110b - Reserved 00_1111b - Reserved 01_0000b - Reserved 01_0001b - Reserved 01_0010b - Reserved 01_0011b - Reserved 01_0100b - TRIG_IN0 input is selected 01_0101b - TRIG_IN1 input is selected 01_0110b - TRIG_IN2 input is selected 01_0111b - TRIG_IN3 input is selected 01_1000b - TRIG_IN4 input is selected 01_1001b - TRIG_IN5 input is selected 01_1010b - TRIG_IN6 input is selected 01_1011b - TRIG_IN7 input is selected 01_1100b - TRIG_IN8 input is selected 01_1101b - TRIG_IN9 input is selected 01_1110b - TRIG_IN10 input is selected 01_1111b - TRIG_IN11 input is selected 10_0000b - GPIO0 Pin Event Trig 0 input is selected 10_0001b - GPIO1 Pin Event Trig 0 input is selected 10_0010b - GPIO2 Pin Event Trig 0 input is selected 10_0011b - GPIO3 Pin Event Trig 0 input is selected 10_0100b - Reserved 10_0101b - ECC Error Interrupt input is selected 10_0110b - WWDT0/1 Interrupt input is selected 10_0111b - Reserved 10_1000b - Reserved 10_1001b - Reserved 10_1010b - Reserved 10_1011b - Reserved

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	10_1100b - Reserved 10_1101b - Reserved 10_1110b - Reserved 10_1111b - Reserved 11_0000b - Reserved 11_0001b - Reserved 11_0010b - Reserved 11_0011b - Reserved 11_0100b - Reserved 11_0101b - Reserved 11_0110b - Reserved 11_0111b - Reserved 11_1000b - Reserved 11_1001b - Reserved 11_1010b - Reserved 11_1011b - Reserved 11_1100b - Reserved 11_1101b - Reserved All other values are reserved.

12.6.1.18 eFlexPWM0 FAULT2 Input Trigger Connection (FlexPWM0_FAULT2)

Offset

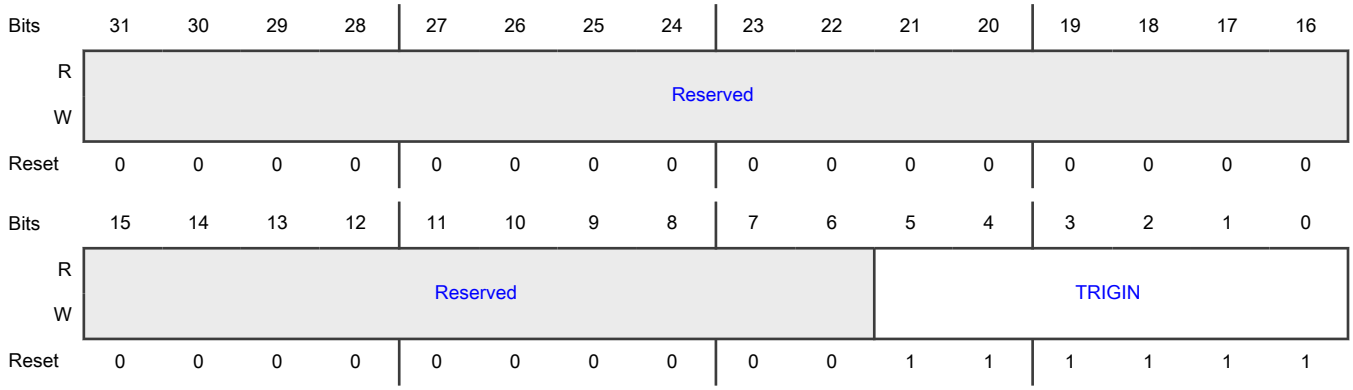
Register	Offset
FlexPWM0_FAULT2	3C8h

Function
Selects the eFlexPWM0 FAULT2 trigger inputs.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram



Fields

Field	Function
31-6 —	Reserved
5-0 TRIGIN	Input Trigger Connection for eFlexPWM0 FAULT2 00_0000b - Reserved 00_0001b - ARM_TXEV input is selected 00_0010b - Reserved 00_0011b - Reserved 00_0100b - Reserved 00_0101b - Reserved 00_0110b - CMP0_OUT input is selected 00_0111b - Reserved 00_1000b - Reserved 00_1001b - CTimer0_MAT2 input is selected 00_1010b - CTimer0_MAT3 input is selected 00_1011b - CTimer1_MAT2 input is selected 00_1100b - CTimer1_MAT3 input is selected 00_1101b - Reserved 00_1110b - Reserved 00_1111b - Reserved 01_0000b - Reserved 01_0001b - Reserved 01_0010b - Reserved 01_0011b - Reserved

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	01_0100b - TRIG_IN0 input is selected 01_0101b - TRIG_IN1 input is selected 01_0110b - TRIG_IN2 input is selected 01_0111b - TRIG_IN3 input is selected 01_1000b - TRIG_IN4 input is selected 01_1001b - TRIG_IN5 input is selected 01_1010b - TRIG_IN6 input is selected 01_1011b - TRIG_IN7 input is selected 01_1100b - TRIG_IN8 input is selected 01_1101b - TRIG_IN9 input is selected 01_1110b - TRIG_IN10 input is selected 01_1111b - TRIG_IN11 input is selected 10_0000b - GPIO0 Pin Event Trig 0 input is selected 10_0001b - GPIO1 Pin Event Trig 0 input is selected 10_0010b - GPIO2 Pin Event Trig 0 input is selected 10_0011b - GPIO3 Pin Event Trig 0 input is selected 10_0100b - Reserved 10_0101b - ECC Error Interrupt input is selected 10_0110b - WWDT0/1 Interrupt input is selected 10_0111b - Reserved 10_1000b - Reserved 10_1001b - Reserved 10_1010b - Reserved 10_1011b - Reserved 10_1100b - Reserved 10_1101b - Reserved 10_1110b - Reserved 10_1111b - Reserved 11_0000b - Reserved 11_0001b - Reserved 11_0010b - Reserved 11_0011b - Reserved 11_0100b - Reserved

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	11_0101b - Reserved 11_0110b - Reserved 11_0111b - Reserved 11_1000b - Reserved 11_1001b - Reserved 11_1010b - Reserved 11_1011b - Reserved 11_1100b - Reserved 11_1101b - Reserved All other values are reserved.

12.6.1.19 eFlexPWM0 FAULT3 Input Trigger Connection (FlexPWM0_FAULT3)

Offset

Register	Offset
FlexPWM0_FAULT3	3CCh

Function

Selects the eFlexPWM0 FAULT3 trigger inputs.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved										TRIGIN					
Reset	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1

Fields

Field	Function
31-6	Reserved

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
—	
5-0 TRIGIN	Input Trigger Connection for eFlexPWM0 FAULT3 00_0000b - Reserved 00_0001b - ARM_TXEV input is selected 00_0010b - Reserved 00_0011b - Reserved 00_0100b - Reserved 00_0101b - Reserved 00_0110b - CMP0_OUT input is selected 00_0111b - Reserved 00_1000b - Reserved 00_1001b - CTimer0_MAT2 input is selected 00_1010b - CTimer0_MAT3 input is selected 00_1011b - CTimer1_MAT2 input is selected 00_1100b - CTimer1_MAT3 input is selected 00_1101b - Reserved 00_1110b - Reserved 00_1111b - Reserved 01_0000b - Reserved 01_0001b - Reserved 01_0010b - Reserved 01_0011b - Reserved 01_0100b - TRIG_IN0 input is selected 01_0101b - TRIG_IN1 input is selected 01_0110b - TRIG_IN2 input is selected 01_0111b - TRIG_IN3 input is selected 01_1000b - TRIG_IN4 input is selected 01_1001b - TRIG_IN5 input is selected 01_1010b - TRIG_IN6 input is selected 01_1011b - TRIG_IN7 input is selected 01_1100b - TRIG_IN8 input is selected 01_1101b - TRIG_IN9 input is selected 01_1110b - TRIG_IN10 input is selected

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	01_1111b - TRIG_IN11 input is selected 10_0000b - GPIO0 Pin Event Trig 0 input is selected 10_0001b - GPIO1 Pin Event Trig 0 input is selected 10_0010b - GPIO2 Pin Event Trig 0 input is selected 10_0011b - GPIO3 Pin Event Trig 0 input is selected 10_0100b - Reserved 10_0101b - ECC Error Interrupt input is selected 10_0110b - WWDT0/1 Interrupt input is selected 10_0111b - Reserved 10_1000b - Reserved 10_1001b - Reserved 10_1010b - Reserved 10_1011b - Reserved 10_1100b - Reserved 10_1101b - Reserved 10_1110b - Reserved 10_1111b - Reserved 11_0000b - Reserved 11_0001b - Reserved 11_0010b - Reserved 11_0011b - Reserved 11_0100b - Reserved 11_0101b - Reserved 11_0110b - Reserved 11_0111b - Reserved 11_1000b - Reserved 11_1001b - Reserved 11_1010b - Reserved 11_1011b - Reserved 11_1100b - Reserved 11_1101b - Reserved All other values are reserved.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

12.6.1.20 eFlexPWM0 EXT_FORCE Input Trigger Connection (FlexPWM0_FORCE)

Offset

Register	Offset
FlexPWM0_FORCE	3D0h

Function

This register selects the eFlexPWM0 EXT_FORCE trigger inputs.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved										TRIGIN					
Reset	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1

Fields

Field	Function
31-6 —	Reserved
5-0 TRIGIN	Input Trigger Connection for eFlexPWM0 EXT_FORCE 00_0000b - Reserved 00_0001b - ARM_TXEV input is selected 00_0010b - Reserved 00_0011b - Reserved 00_0100b - Reserved 00_0101b - Reserved 00_0110b - CMP0_OUT input is selected 00_0111b - Reserved 00_1000b - Reserved 00_1001b - CTimer0_MAT2 input is selected 00_1010b - CTimer0_MAT3 input is selected

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	00_1011b - CTimer1_MAT2 input is selected 00_1100b - CTimer1_MAT3 input is selected 00_1101b - Reserved 00_1110b - Reserved 00_1111b - Reserved 01_0000b - Reserved 01_0001b - Reserved 01_0010b - Reserved 01_0011b - Reserved 01_0100b - TRIG_IN0 input is selected 01_0101b - TRIG_IN1 input is selected 01_0110b - TRIG_IN2 input is selected 01_0111b - TRIG_IN3 input is selected 01_1000b - TRIG_IN4 input is selected 01_1001b - TRIG_IN5 input is selected 01_1010b - TRIG_IN6 input is selected 01_1011b - TRIG_IN7 input is selected 01_1100b - TRIG_IN8 input is selected 01_1101b - TRIG_IN9 input is selected 01_1110b - TRIG_IN10 input is selected 01_1111b - TRIG_IN11 input is selected 10_0000b - GPIO0 Pin Event Trig 0 input is selected 10_0001b - GPIO1 Pin Event Trig 0 input is selected 10_0010b - GPIO2 Pin Event Trig 0 input is selected 10_0011b - GPIO3 Pin Event Trig 0 input is selected 10_0100b - Reserved 10_0101b - ECC Error Interrupt input is selected 10_0110b - WWDT0/1 Interrupt input is selected 10_0111b - Reserved 10_1000b - Reserved 10_1001b - Reserved 10_1010b - Reserved 10_1011b - Reserved

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	10_1100b - Reserved 10_1101b - Reserved 10_1110b - Reserved 10_1111b - Reserved 11_0000b - Reserved 11_0001b - Reserved 11_0010b - Reserved 11_0011b - Reserved 11_0100b - Reserved 11_0101b - Reserved 11_0110b - Reserved 11_0111b - Reserved 11_1000b - Reserved 11_1001b - Reserved 11_1010b - Reserved 11_1011b - Reserved 11_1100b - Reserved 11_1101b - Reserved All other values are reserved.

12.6.1.21 eFlexPWM0 EXT_CLK Trigger (PWM0_EXT_CLK)

Offset

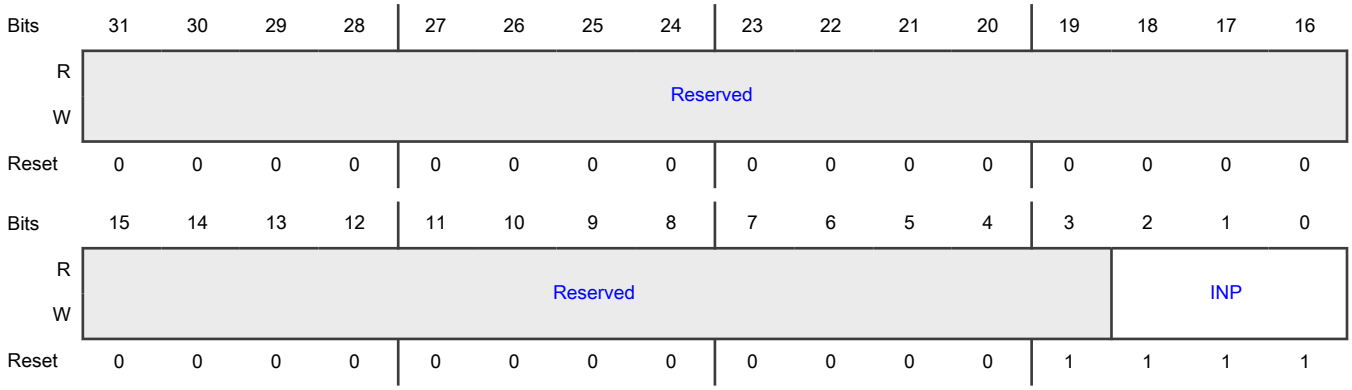
Register	Offset
PWM0_EXT_CLK	420h

Function
eFlexPWM0 EXT_CLK trigger connections

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram



Fields

Field	Function
31-3 —	Reserved
2-0 INP	Input Trigger Connection for eFlexPWM0 EXT_CLK 000b - Reserved 001b - clk_16k[1] input is selected 010b - clk_in input is selected 011b - Reserved 100b - Reserved 101b - TRIG_IN0 input is selected 110b - TRIG_IN7 input is selected 111b - Reserved All other values are reserved.

12.6.1.22 External Trigger a Connections (EXT_TRIG0 - EXT_TRIG7)

Offset

Register	Offset
EXT_TRIG0	4C0h
EXT_TRIG1	4C4h
EXT_TRIG2	4C8h
EXT_TRIG3	4CCh
EXT_TRIG4	4D0h
EXT_TRIG6	4D8h

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

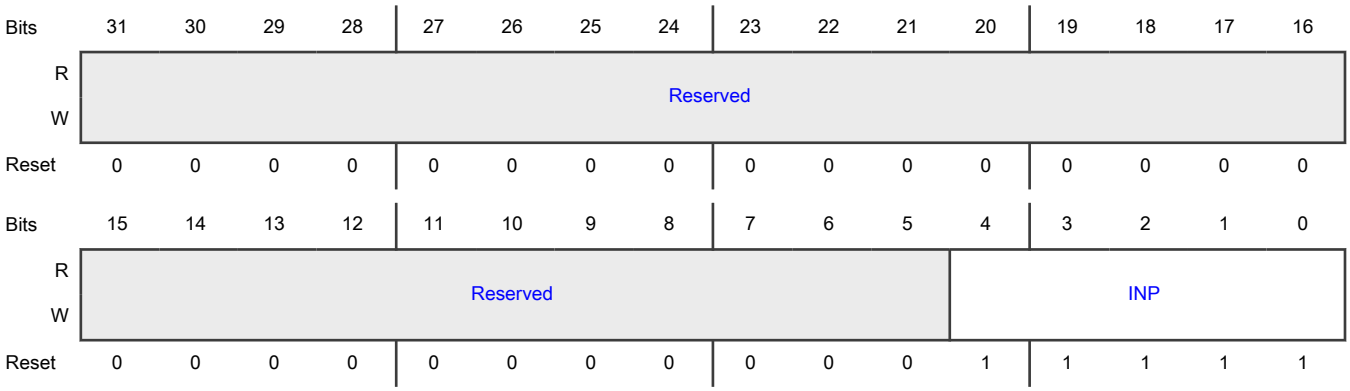
Table continued from the previous page...

Register	Offset
EXT_TRIG7	4DCh

Function

Selects the external trigger a inputs.

Diagram



Fields

Field	Function
31-5 —	Reserved
4-0 INP	External Trigger Input Connections 0_0000b - Reserved 0_0001b - Reserved 0_0010b - Reserved 0_0011b - Reserved 0_0100b - Reserved 0_0101b - Reserved 0_0110b - CMP0_OUT input is selected 0_0111b - Reserved 0_1000b - Reserved 0_1001b - LPUART0 ipp_do_lpuart_txd input is selected 0_1010b - LPUART1 ipp_do_lpuart_txd input is selected 0_1011b - LPUART2 ipp_do_lpuart_txd input is selected 0_1100b - LPUART3 ipp_do_lpuart_txd input is selected

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	0_1101b - Reserved 0_1110b - Reserved 0_1111b - Reserved 1_0000b - Reserved 1_0001b - Reserved 1_0010b - Reserved 1_0011b - RTC_1Hz_CLK input is selected All other values are reserved.

12.6.1.23 LPI2C0 Trigger Input Connections (LPI2C0_TRIG)

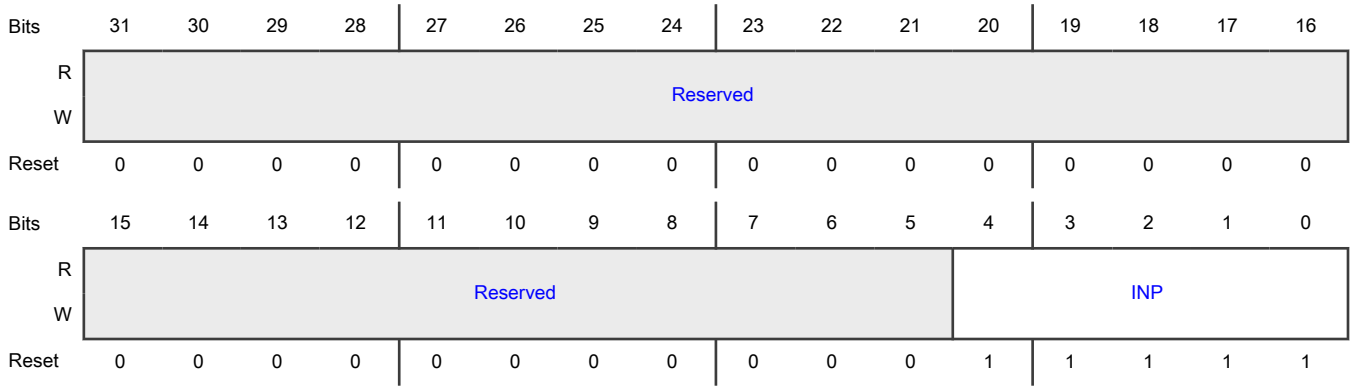
Offset

Register	Offset
LPI2C0_TRIG	5A0h

Function

Selects LPI2C0 trigger inputs.

Diagram



Fields

Field	Function
31-5 —	Reserved
4-0	LPI2C0 Trigger Input Connections

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
INP	0_0000b - Reserved 0_0001b - Reserved 0_0010b - Reserved 0_0011b - Reserved 0_0100b - Reserved 0_0101b - Reserved 0_0110b - CMP0_OUT input is selected 0_0111b - Reserved 0_1000b - Reserved 0_1001b - CTimer0_MAT0 input is selected 0_1010b - CTimer0_MAT1 input is selected 0_1011b - CTimer1_MAT0 input is selected 0_1100b - CTimer1_MAT1 input is selected 0_1101b - Reserved 0_1110b - Reserved 0_1111b - LPTMR0 input is selected 1_0000b - Reserved 1_0001b - TRIG_IN0 input is selected 1_0010b - TRIG_IN1 input is selected 1_0011b - TRIG_IN2 input is selected 1_0100b - TRIG_IN3 input is selected 1_0101b - TRIG_IN4 input is selected 1_0110b - TRIG_IN5 input is selected 1_0111b - TRIG_IN6 input is selected 1_1000b - TRIG_IN7 input is selected 1_1001b - GPIO0 Pin Event Trig 0 input is selected 1_1010b - GPIO1 Pin Event Trig 0 input is selected 1_1011b - GPIO2 Pin Event Trig 0 input is selected 1_1100b - GPIO3 Pin Event Trig 0 input is selected 1_1101b - Reserved 1_1110b - WUU input is selected All other values are reserved.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

12.6.1.24 LPSPI0 Trigger Input Connections (LPSPI0_TRIG)

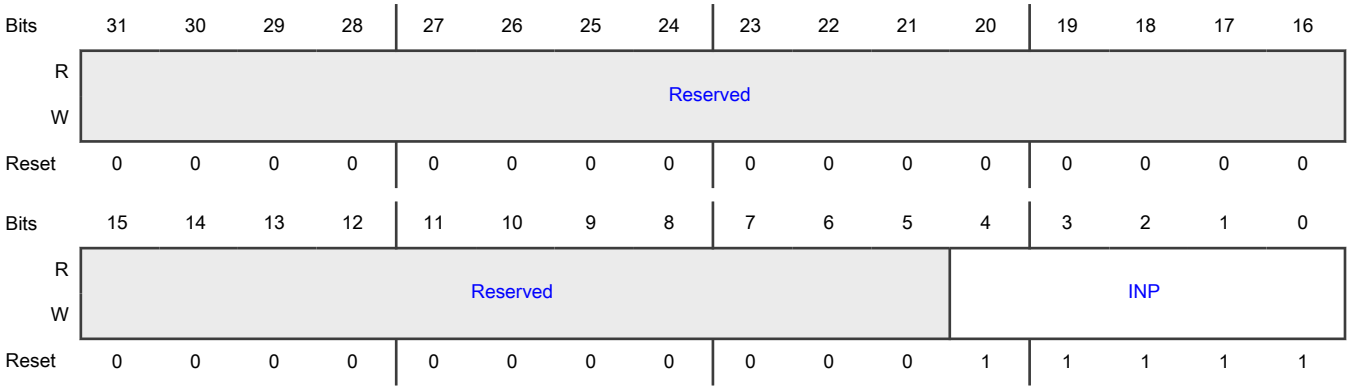
Offset

Register	Offset
LPSPI0_TRIG	5E0h

Function

Selects the LPSPI0 trigger inputs.

Diagram



Fields

Field	Function
31-5 —	Reserved
4-0 INP	LPSPI0 Trigger Input Connections 0_0000b - Reserved 0_0001b - Reserved 0_0010b - Reserved 0_0011b - Reserved 0_0100b - Reserved 0_0101b - Reserved 0_0110b - CMP0_OUT input is selected 0_0111b - Reserved 0_1000b - Reserved 0_1001b - CTimer0_MAT1 input is selected 0_1010b - CTimer0_MAT2 input is selected

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	0_1011b - CTimer1_MAT1 input is selected 0_1100b - CTimer1_MAT2 input is selected 0_1101b - Reserved 0_1110b - Reserved 0_1111b - LPTMR0 input is selected 1_0000b - Reserved 1_0001b - TRIG_IN0 input is selected 1_0010b - TRIG_IN1 input is selected 1_0011b - TRIG_IN2 input is selected 1_0100b - TRIG_IN3 input is selected 1_0101b - TRIG_IN4 input is selected 1_0110b - TRIG_IN5 input is selected 1_0111b - TRIG_IN6 input is selected 1_1000b - TRIG_IN7 input is selected 1_1001b - GPIO0 Pin Event Trig 0 input is selected 1_1010b - GPIO1 Pin Event Trig 0 input is selected 1_1011b - GPIO2 Pin Event Trig 0 input is selected 1_1100b - GPIO3 Pin Event Trig 0 input is selected 1_1101b - Reserved 1_1110b - WUU input is selected All other values are reserved.

12.6.1.25 LPUART0 Trigger Input Connections (LPUART0)

Offset

Register	Offset
LPUART0	620h

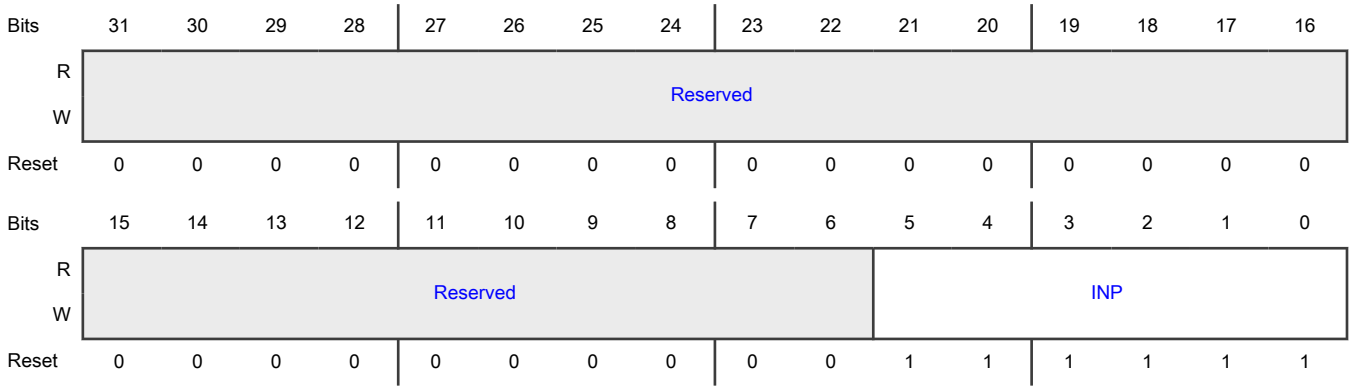
Function

Selects the LPUART0 trigger inputs.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram



Fields

Field	Function
31-6 —	Reserved
5-0 INP	LPUART0 Trigger Input Connections 00_0000b - Reserved 00_0001b - Reserved 00_0010b - Reserved 00_0011b - Reserved 00_0100b - Reserved 00_0101b - Reserved 00_0110b - CMP0_OUT input is selected 00_0111b - Reserved 00_1000b - Reserved 00_1001b - CTimer0_MAT2 input is selected 00_1010b - CTimer0_MAT3 input is selected 00_1011b - CTimer1_MAT2 input is selected 00_1100b - CTimer1_MAT3 input is selected 00_1101b - Reserved 00_1110b - Reserved 00_1111b - LPTMR0 input is selected 01_0000b - Reserved 01_0001b - TRIG_IN0 input is selected 01_0010b - TRIG_IN1 input is selected 01_0011b - TRIG_IN2 input is selected

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	01_0100b - TRIG_IN3 input is selected 01_0101b - TRIG_IN4 input is selected 01_0110b - TRIG_IN5 input is selected 01_0111b - TRIG_IN6 input is selected 01_1000b - TRIG_IN7 input is selected 01_1001b - TRIG_IN8 input is selected 01_1010b - TRIG_IN9 input is selected 01_1011b - TRIG_IN10 input is selected 01_1100b - TRIG_IN11 input is selected 01_1101b - GPIO0 Pin Event Trig 0 input is selected 01_1110b - GPIO1 Pin Event Trig 0 input is selected 01_1111b - GPIO2 Pin Event Trig 0 input is selected 10_0000b - GPIO3 Pin Event Trig 0 input is selected 10_0001b - Reserved 10_0010b - WUU input is selected 10_0011b - Reserved 10_0100b - Reserved 10_0101b - Reserved 10_0110b - Reserved 10_0111b - Reserved 10_1000b - Reserved 10_1001b - Reserved 10_1010b - Reserved 10_1011b - Reserved 10_1100b - Reserved 10_1101b - Reserved 10_1110b - Reserved 10_1111b - Reserved All other values are reserved.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

12.6.1.26 LPUART1 Trigger Input Connections (LPUART1)

Offset

Register	Offset
LPUART1	640h

Function
Selects the LPUART1 trigger inputs.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved										INP					
Reset	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1

Fields

Field	Function
31-6 —	Reserved
5-0 INP	LPUART1 Trigger Input Connections 00_0000b - Reserved 00_0001b - Reserved 00_0010b - Reserved 00_0011b - Reserved 00_0100b - Reserved 00_0101b - Reserved 00_0110b - CMP0_OUT input is selected 00_0111b - Reserved 00_1000b - Reserved 00_1001b - CTimer0_MAT2 input is selected 00_1010b - CTimer0_MAT3 input is selected

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	00_1011b - CTimer1_MAT2 input is selected 00_1100b - CTimer1_MAT3 input is selected 00_1101b - Reserved 00_1110b - Reserved 00_1111b - LPTMR0 input is selected 01_0000b - Reserved 01_0001b - TRIG_IN0 input is selected 01_0010b - TRIG_IN1 input is selected 01_0011b - TRIG_IN2 input is selected 01_0100b - TRIG_IN3 input is selected 01_0101b - TRIG_IN4 input is selected 01_0110b - TRIG_IN5 input is selected 01_0111b - TRIG_IN6 input is selected 01_1000b - TRIG_IN7 input is selected 01_1001b - TRIG_IN8 input is selected 01_1010b - TRIG_IN9 input is selected 01_1011b - TRIG_IN10 input is selected 01_1100b - TRIG_IN11 input is selected 01_1101b - GPIO0 Pin Event Trig 0 input is selected 01_1110b - GPIO1 Pin Event Trig 0 input is selected 01_1111b - GPIO2 Pin Event Trig 0 input is selected 10_0000b - GPIO3 Pin Event Trig 0 input is selected 10_0001b - Reserved 10_0010b - WUU input is selected 10_0011b - Reserved 10_0100b - Reserved 10_0101b - Reserved 10_0110b - Reserved 10_0111b - Reserved 10_1000b - Reserved 10_1001b - Reserved 10_1010b - Reserved 10_1011b - Reserved

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	10_1100b - Reserved 10_1101b - Reserved 10_1110b - Reserved 10_1111b - Reserved All other values are reserved.

12.6.1.27 LPUART2 Trigger Input Connections (LPUART2)

Offset

Register	Offset
LPUART2	660h

Function

Selects the LPUART2 trigger inputs.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved										INP					
Reset	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1

Fields

Field	Function
31-6 —	Reserved
5-0 INP	LPUART2 Trigger Input Connections 00_0000b - Reserved 00_0001b - Reserved

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	00_0010b - Reserved 00_0011b - Reserved 00_0100b - Reserved 00_0101b - Reserved 00_0110b - CMP0_OUT input is selected 00_0111b - Reserved 00_1000b - Reserved 00_1001b - CTimer0_MAT2 input is selected 00_1010b - CTimer0_MAT3 input is selected 00_1011b - CTimer1_MAT2 input is selected 00_1100b - CTimer1_MAT3 input is selected 00_1101b - Reserved 00_1110b - Reserved 00_1111b - LPTMR0 input is selected 01_0000b - Reserved 01_0001b - TRIG_IN0 input is selected 01_0010b - TRIG_IN1 input is selected 01_0011b - TRIG_IN2 input is selected 01_0100b - TRIG_IN3 input is selected 01_0101b - TRIG_IN4 input is selected 01_0110b - TRIG_IN5 input is selected 01_0111b - TRIG_IN6 input is selected 01_1000b - TRIG_IN7 input is selected 01_1001b - TRIG_IN8 input is selected 01_1010b - TRIG_IN9 input is selected 01_1011b - TRIG_IN10 input is selected 01_1100b - TRIG_IN11 input is selected 01_1101b - GPIO0 Pin Event Trig 0 input is selected 01_1110b - GPIO1 Pin Event Trig 0 input is selected 01_1111b - GPIO2 Pin Event Trig 0 input is selected 10_0000b - GPIO3 Pin Event Trig 0 input is selected 10_0001b - Reserved 10_0010b - WUU input is selected

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	10_0011b - Reserved 10_0100b - Reserved 10_0101b - Reserved 10_0110b - Reserved 10_0111b - Reserved 10_1000b - Reserved 10_1001b - Reserved 10_1010b - Reserved 10_1011b - Reserved 10_1100b - Reserved 10_1101b - Reserved 10_1110b - Reserved 10_1111b - Reserved All other values are reserved.

12.6.1.28 LPUART3 Trigger Input Connections (LPUART3)

Offset

Register	Offset
LPUART3	680h

Function

Selects the LPUART3 trigger inputs.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved										INP					
Reset	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Fields

Field	Function
31-6 —	Reserved
5-0 INP	LPUART3 Trigger Input Connections 00_0000b - Reserved 00_0001b - Reserved 00_0010b - Reserved 00_0011b - Reserved 00_0100b - Reserved 00_0101b - Reserved 00_0110b - CMP0_OUT input is selected 00_0111b - Reserved 00_1000b - Reserved 00_1001b - CTimer0_MAT2 input is selected 00_1010b - CTimer0_MAT3 input is selected 00_1011b - CTimer1_MAT2 input is selected 00_1100b - CTimer1_MAT3 input is selected 00_1101b - Reserved 00_1110b - Reserved 00_1111b - LPTMR0 input is selected 01_0000b - Reserved 01_0001b - TRIG_IN0 input is selected 01_0010b - TRIG_IN1 input is selected 01_0011b - TRIG_IN2 input is selected 01_0100b - TRIG_IN3 input is selected 01_0101b - TRIG_IN4 input is selected 01_0110b - TRIG_IN5 input is selected 01_0111b - TRIG_IN6 input is selected 01_1000b - TRIG_IN7 input is selected 01_1001b - TRIG_IN8 input is selected 01_1010b - TRIG_IN9 input is selected 01_1011b - TRIG_IN10 input is selected 01_1100b - TRIG_IN11 input is selected 01_1101b - GPIO0 Pin Event Trig 0 input is selected

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	01_1110b - GPIO1 Pin Event Trig 0 input is selected 01_1111b - GPIO2 Pin Event Trig 0 input is selected 10_0000b - GPIO3 Pin Event Trig 0 input is selected 10_0001b - Reserved 10_0010b - WUU input is selected 10_0011b - Reserved 10_0100b - Reserved 10_0101b - Reserved 10_0110b - Reserved 10_0111b - Reserved 10_1000b - Reserved 10_1001b - Reserved 10_1010b - Reserved 10_1011b - Reserved 10_1100b - Reserved 10_1101b - Reserved 10_1110b - Reserved 10_1111b - Reserved All other values are reserved.

12.6.1.29 Trigger Filter Prescaller (TRIGFIL_PRSC)

Offset

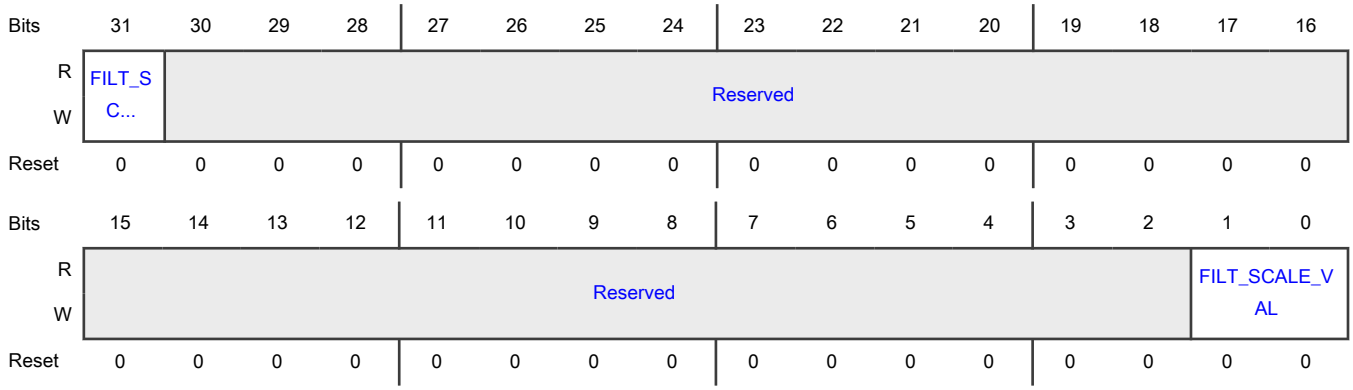
Register	Offset
TRIGFIL_PRSC	A00h

Function
Configures trigger filter prescaller

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram



Fields

Field	Function
31 FILT_SCALE_EN	Trigger Filter Prescaller Enable Enables trigger filter prescaller. 0b - Disable prescaller 1b - Enabled prescaller
30-2 —	Reserved
1-0 FILT_SCALE_VAL	Filter Prescaller Value 00b - Bypass the clock 01b - Divide 2 10b - Divide 4 11b - Divide 8

12.6.1.30 Trigger Filter Status (TRIGFIL_STAT0)

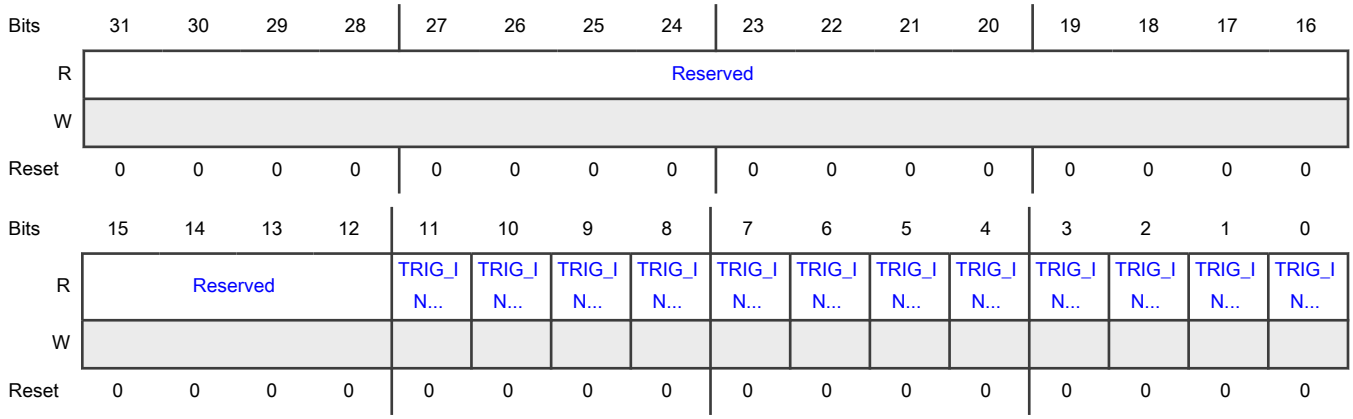
Offset

Register	Offset
TRIGFIL_STAT0	A04h

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram



Fields

Field	Function
31-12 —	Reserved
11 TRIG_IN11_VAL	TRIG_IN value 0b - TRIG_IN11 is 0 1b - TRIG_IN11 is 1
10 TRIG_IN10_VAL	TRIG_IN value 0b - TRIG_IN10 is 0 1b - TRIG_IN10 is 1
9 TRIG_IN9_VAL	TRIG_IN value 0b - TRIG_IN9 is 0 1b - TRIG_IN9 is 1
8 TRIG_IN8_VAL	TRIG_IN value 0b - TRIG_IN8 is 0 1b - TRIG_IN8 is 1
7 TRIG_IN7_VAL	TRIG_IN value 0b - TRIG_IN7 is 0 1b - TRIG_IN7 is 1
6 TRIG_IN6_VAL	TRIG_IN value 0b - TRIG_IN6 is 0 1b - TRIG_IN6 is 1
5	TRIG_IN value

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
TRIG_IN5_VAL	0b - TRIG_IN5 is 0 1b - TRIG_IN5 is 1
4 TRIG_IN4_VAL	TRIG_IN value 0b - TRIG_IN4 is 0 1b - TRIG_IN4 is 1
3 TRIG_IN3_VAL	TRIG_IN value 0b - TRIG_IN3 is 0 1b - TRIG_IN3 is 1
2 TRIG_IN2_VAL	TRIG_IN value 0b - TRIG_IN2 is 0 1b - TRIG_IN2 is 1
1 TRIG_IN1_VAL	TRIG_IN value 0b - TRIG_IN1 is 0 1b - TRIG_IN1 is 1
0 TRIG_IN0_VAL	TRIG_IN value 0b - TRIG_IN0 is 0 1b - TRIG_IN0 is 1

12.6.1.31 TRIGFIL Control (TRIGFIL0 - TRIGFIL11)

Offset

For a = 0 to 11:

Register	Offset
TRIGFILa	A10h + (a × 4h)

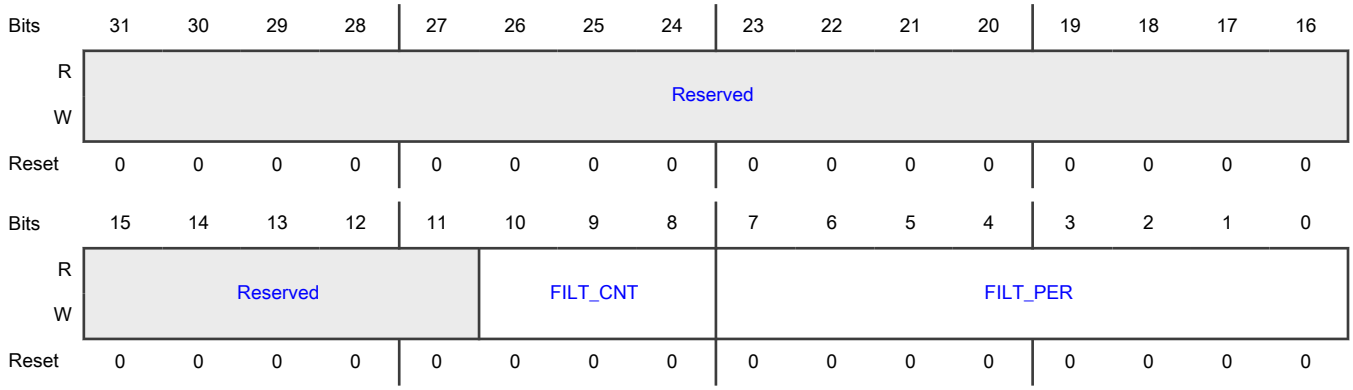
Function

Controls the trigger filter.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram



Fields

Field	Function
31-11 —	Reserved
10-8 FILT_CNT	Input Filter Sample Count Configures the number of consecutive samples that must agree prior to the input filter accepting an input transition. The number of samples is the decimal value of this field plus three: the bit field value of 0-7 represents 3-10 samples, respectively. The value of FILT_CNT affects the input latency.
7-0 FILT_PER	Input Filter Sample Period Configures the sampling period (in IPBus clock cycles) of the fault pin input filter. Each input is sampled multiple times at the rate specified by this field. If FILT_PER is 0x00, then the input filter is bypassed. The value of FILT_PER affects the input latency. When changing values for FILT_PER from one non-zero value to another non-zero value, first write a value of zero to clear the filter.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Chapter 13
System Controller (SYSCON)

13.1 Chip-specific SYSCON information

Table 73. Reference links to related information

Topic	Related module	Reference
Full description	SYSCON	SYSCON
System memory map		Memory map
Clocking		Clock distribution
Power Management		Power Management
Signal multiplexing	Port control	Signal multiplexing

13.1.1 Module instances

This chip has one instance of SYSCON module.

13.1.2 Configuration

Configure the SYSCON block as follows:

- No clock configuration is needed. The clock to the SYSCON block is always enabled.
- The SYSCON block controls use of the CLKOUT pin, which must also be configured through Port control module.

13.2 Overview

The SYSCON module provides controls and configurations on the system and peripherals for the multiple functions.

13.2.1 Features

The SYSCON module supports the following features and configurations:

- System and bus configuration
  - AHB matrix priority
  - CPU control and status
  - CPU debug access
  - NMI source select
  - Calibrate system tick timer
  - Boot control
- System/Peripherals - clock select and control
  - Allows enabling and selection of clocks to individual peripherals and memories
  - Allows configuration of clock dividers
- Memory configuration
  - RAM ECC
  - FLASH cache, bus error

© 2026 NXP B.V. All rights reserved.

- Peripherals configuration
  - PWM control
  - CTIMER global start
- Security configuration
  - Security control
  - Security attestation
- Reset control
  - Monitors and release resets to individual peripherals
- Device ID register

13.3 Functional description

The MRCC submodule provides on-chip modules their own dedicated MRCC bits for clock gating, reset control and configuration options. Generally, MRCC register contains a clock configuration (CC = {MRCC_GLB_ACC*[peripherals name]*, MRCC_GLB_CC*[peripherals name]*}) field for the module's clocks.

NOTE

Before a module can be used, its clocks must be enabled (CC != 00) and it must be released from reset (MRCC_GLB_RST*[peripherals name]*= 1). If a module is *not* released from reset (MRCC_GLB_RST*[peripherals name]*= 0), an attempt to access a register within that module is terminated with a bus error.

If a module has a functional clock, its MRCC register may provide options for the clock source, selected by programming the mux select (MUX) field. The selected functional clock source can then be divided by programming the divider (DIV) field.

NOTE

Before configuring a functional clock, the module's clocks must be disabled (CC = 0b00).

13.3.1 Signals

Table 74. SYSCON pin description

Function	Type	Pin	Description
CLKOUT	O	P0_6, P3_8	The CLKOUT function provides a clock output through these pins if they are available in the package.

13.4 Memory map and register definition

This section includes the SYSCON module memory map and detailed descriptions of all registers.

13.4.1 MRCC register descriptions

13.4.1.1 MRCC memory map

SYSCON__MRCC base address: 4009_1800h

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Offset	Register	Width (In bits)	Access	Reset value
0h	Peripheral Reset Control 0 (MRCC_GLB_RST0)	32	RW	0000_0000h
4h	Peripheral Reset Control Set 0 (MRCC_GLB_RST0_SET)	32	W	0000_0000h
8h	Peripheral Reset Control Clear 0 (MRCC_GLB_RST0_CLR)	32	W	0000_0000h
10h	Peripheral Reset Control 1 (MRCC_GLB_RST1)	32	RW	0000_0000h
14h	Peripheral Reset Control Set 1 (MRCC_GLB_RST1_SET)	32	W	0000_0000h
18h	Peripheral Reset Control Clear 1 (MRCC_GLB_RST1_CLR)	32	W	0000_0000h
20h	Peripheral Reset Control 2 (MRCC_GLB_RST2)	32	RW	0000_0000h
24h	Peripheral Reset Control Set 2 (MRCC_GLB_RST2_SET)	32	W	0000_0000h
28h	Peripheral Reset Control Clear 2 (MRCC_GLB_RST2_CLR)	32	W	0000_0000h
60h	AHB Clock Control 0 (MRCC_GLB_CC0)	32	RW	0000_0000h
64h	AHB Clock Control Set 0 (MRCC_GLB_CC0_SET)	32	W	0000_0000h
68h	AHB Clock Control Clear 0 (MRCC_GLB_CC0_CLR)	32	W	0000_0000h
70h	AHB Clock Control 1 (MRCC_GLB_CC1)	32	RW	0000_0000h
74h	AHB Clock Control Set 1 (MRCC_GLB_CC1_SET)	32	W	0000_0000h
78h	AHB Clock Control Clear 1 (MRCC_GLB_CC1_CLR)	32	W	0000_0000h
80h	AHB Clock Control 2 (MRCC_GLB_CC2)	32	RW	0000_0000h
84h	AHB Clock Control Set 2 (MRCC_GLB_CC2_SET)	32	W	0000_0000h
88h	AHB Clock Control Clear 2 (MRCC_GLB_CC2_CLR)	32	W	0000_0000h
C0h	Control Automatic Clock Gating 0 (MRCC_GLB_ACC0)	32	RW	0000_0300h
C4h	Control Automatic Clock Gating 1 (MRCC_GLB_ACC1)	32	RW	0000_0000h
C8h	Control Automatic Clock Gating 2 (MRCC_GLB_ACC2)	32	RW	0000_1000h
100h	CTIMER0 clock selection control (MRCC_CTIMER0_CLKSEL)	32	RW	0000_0007h
104h	CTIMER0 clock divider control (MRCC_CTIMER0_CLKDIV)	32	RW	4000_0000h
108h	CTIMER1 clock selection control (MRCC_CTIMER1_CLKSEL)	32	RW	0000_0007h
10Ch	CTIMER1 clock divider control (MRCC_CTIMER1_CLKDIV)	32	RW	4000_0000h
114h	WWDT0 clock divider control (MRCC_WWDT0_CLKDIV)	32	RW	0000_0000h
118h	WWDT1 clock selection control (MRCC_WWDT1_CLKSEL)	32	RW	0000_0000h
11Ch	WWDT1 clock divider control (MRCC_WWDT1_CLKDIV)	32	RW	0000_0000h
120h	LPI2C0 clock selection control (MRCC_LPI2C0_CLKSEL)	32	RW	0000_0007h
124h	LPI2C0 clock divider control (MRCC_LPI2C0_CLKDIV)	32	RW	4000_0000h
128h	LPSPI0 clock selection control (MRCC_LPSPI0_CLKSEL)	32	RW	0000_0007h
12Ch	LPSPI0 clock divider control (MRCC_LPSPI0_CLKDIV)	32	RW	4000_0000h

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Offset	Register	Width (In bits)	Access	Reset value
130h	LPUART0 clock selection control (MRCC_LPUART0_CLKSEL)	32	RW	0000_0007h
134h	LPUART0 clock divider control (MRCC_LPUART0_CLKDIV)	32	RW	4000_0000h
138h	LPUART1 clock selection control (MRCC_LPUART1_CLKSEL)	32	RW	0000_0007h
13Ch	LPUART1 clock divider control (MRCC_LPUART1_CLKDIV)	32	RW	4000_0000h
140h	LPUART2 clock selection control (MRCC_LPUART2_CLKSEL)	32	RW	0000_0007h
144h	LPUART2 clock divider control (MRCC_LPUART2_CLKDIV)	32	RW	4000_0000h
148h	LPUART3 clock selection control (MRCC_LPUART3_CLKSEL)	32	RW	0000_0007h
14Ch	LPUART3 clock divider control (MRCC_LPUART3_CLKDIV)	32	RW	4000_0000h
150h	LPTMR0 clock selection control (MRCC_LPTMR0_CLKSEL)	32	RW	0000_0007h
154h	LPTMR0 clock divider control (MRCC_LPTMR0_CLKDIV)	32	RW	4000_0000h
158h	ADC clock selection control (MRCC_ADC_CLKSEL)	32	RW	0000_0007h
15Ch	ADC clock divider control (MRCC_ADC_CLKDIV)	32	RW	4000_0000h
164h	CMP0_FUNC clock divider control (MRCC_CMP0_FUNC_CLKDIV)	32	RW	4000_0000h
168h	CMP0_RR clock selection control (MRCC_CMP0_RR_CLKSEL)	32	RW	0000_0007h
16Ch	CMP0_RR clock divider control (MRCC_CMP0_RR_CLKDIV)	32	RW	4000_0000h
170h	DBG_TRACE clock selection control (MRCC_DBG_TRACE_CLKSEL)	32	RW	0000_0000h
174h	DBG_TRACE clock divider control (MRCC_DBG_TRACE_CLKDIV)	32	RW	0000_0000h
178h	CLKOUT clock selection control (MRCC_CLKOUT_CLKSEL)	32	RW	0000_0007h
17Ch	CLKOUT clock divider control (MRCC_CLKOUT_CLKDIV)	32	RW	4000_0000h
180h	SYSTICK clock selection control (MRCC_SYSTICK_CLKSEL)	32	RW	0000_0003h
184h	SYSTICK clock divider control (MRCC_SYSTICK_CLKDIV)	32	RW	4000_0000h

13.4.1.2 Peripheral Reset Control 0 (MRCC_GLB_RST0)

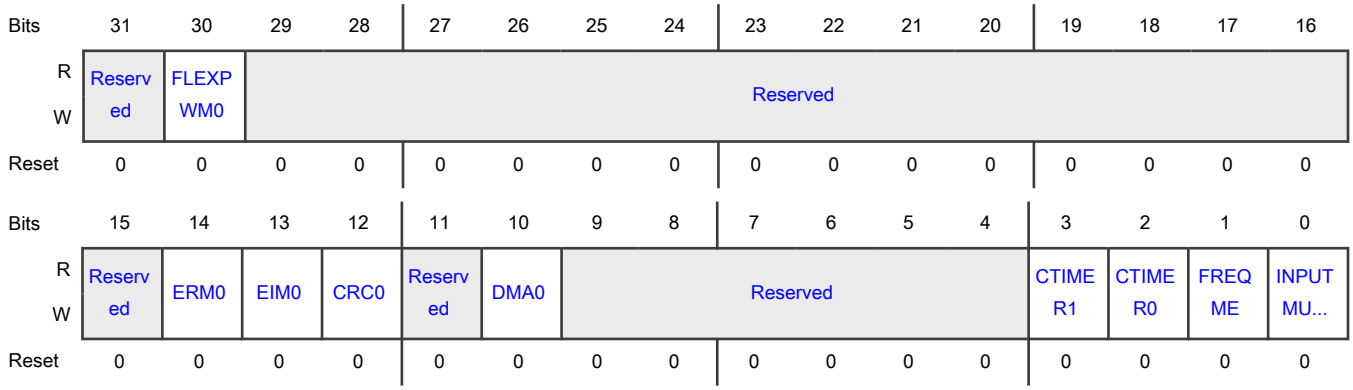
Offset

Register	Offset
MRCC_GLB_RST0	0h

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram



Fields

Field	Function
31 —	reserved reserved
30 FLEXPWM0	FLEXPWM0 0b - Peripheral is held in reset 1b - Peripheral is released from reset
29-15 —	reserved reserved
14 ERM0	ERM0 0b - Peripheral is held in reset 1b - Peripheral is released from reset
13 EIM0	EIM0 0b - Peripheral is held in reset 1b - Peripheral is released from reset
12 CRC0	CRC0 0b - Peripheral is held in reset 1b - Peripheral is released from reset
11 —	reserved reserved
10 DMA0	DMA0 0b - Peripheral is held in reset 1b - Peripheral is released from reset
9-4	reserved

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
—	reserved
3 CTIMER1	CTIMER1 0b - Peripheral is held in reset 1b - Peripheral is released from reset
2 CTIMER0	CTIMER0 0b - Peripheral is held in reset 1b - Peripheral is released from reset
1 FREQME	FREQME 0b - Peripheral is held in reset 1b - Peripheral is released from reset
0 INPUTMUX0	INPUTMUX0 0b - Peripheral is held in reset 1b - Peripheral is released from reset

13.4.1.3 Peripheral Reset Control Set n (MRCC_GLB_RST0_SET - MRCC_GLB_RST2_SET)

Offset

Register	Offset
MRCC_GLB_RST0_SET	4h
MRCC_GLB_RST1_SET	14h
MRCC_GLB_RST2_SET	24h

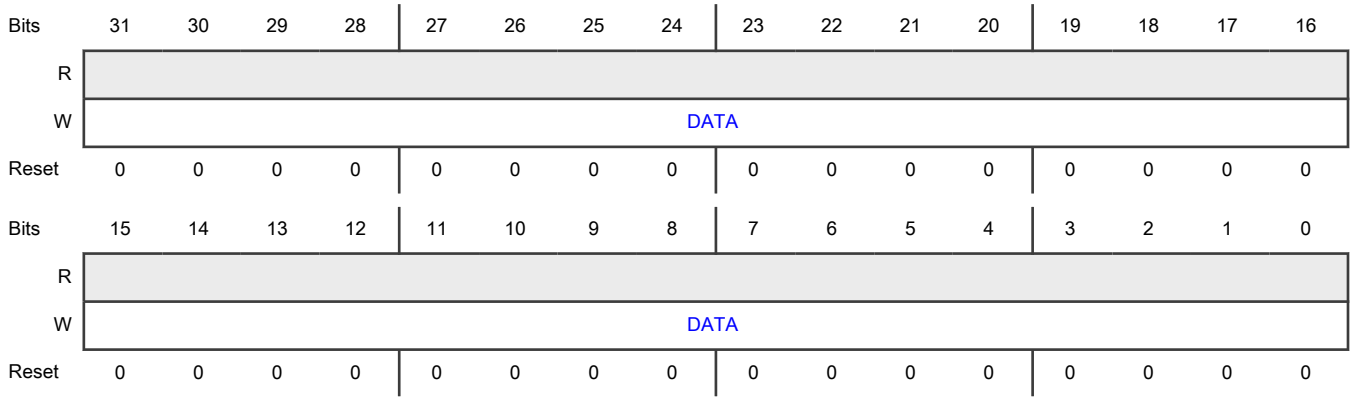
Function

Writing a 1 to a bit position in a write-only MRCC_GLB_RSTn_SET register sets the corresponding position in MRCC_GLB_RSTn.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram



Fields

Field	Function
31-0 DATA	Data array value, refer to corresponding position in MRCC_GLB_RSTn.

13.4.1.4 Peripheral Reset Control Clear n (MRCC_GLB_RST0_CLR - MRCC_GLB_RST2_CLR)

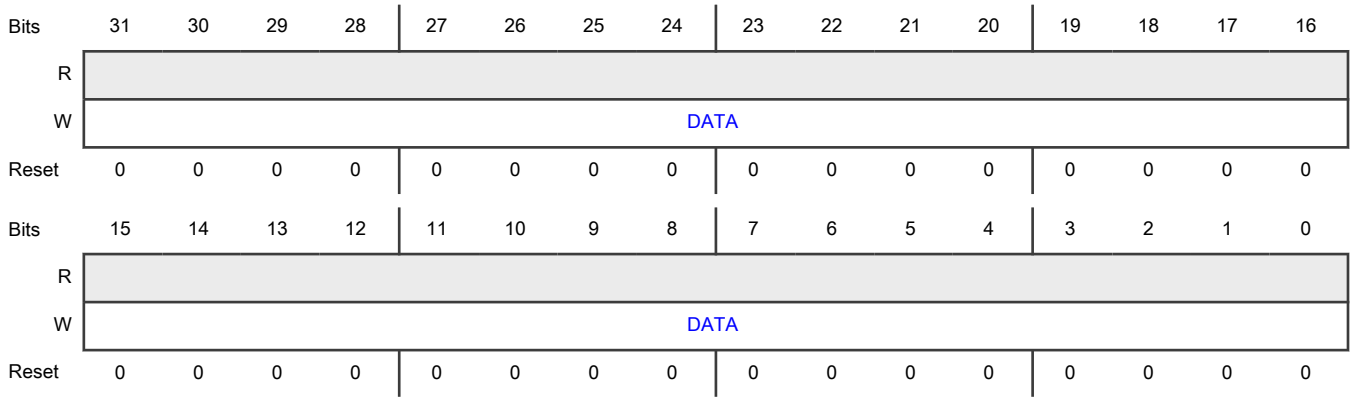
Offset

Register	Offset
MRCC_GLB_RST0_CLR	8h
MRCC_GLB_RST1_CLR	18h
MRCC_GLB_RST2_CLR	28h

Function

Writing a 1 to a bit position in a write-only MRCC_GLB_RSTn_CLR register clears the corresponding position in MRCC_GLB_RSTn.

Diagram



All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Fields

Field	Function
31-0 DATA	Data array value, refer to corresponding position in MRCC_GLB_RSTn.

13.4.1.5 Peripheral Reset Control 1 (MRCC_GLB_RST1)

Offset

Register	Offset
MRCC_GLB_RST1	10h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved						OPAMP0		Reserved			CMP0	Reserved			ADC0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved				LPUART3	LPUART2	LPUART1	LPUART0	Reserved			LPSPI0	Reserved			LPI2C0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-26 —	reserved reserved
25 OPAMP0	OPAMP0 0b - Peripheral is held in reset 1b - Peripheral is released from reset
24-21 —	reserved reserved
20 CMP0	CMP0 0b - Peripheral is held in reset 1b - Peripheral is released from reset

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
19-17 —	reserved reserved
16 ADC0	ADC0 0b - Peripheral is held in reset 1b - Peripheral is released from reset
15-12 —	reserved reserved
11 LPUART3	LPUART3 0b - Peripheral is held in reset 1b - Peripheral is released from reset
10 LPUART2	LPUART2 0b - Peripheral is held in reset 1b - Peripheral is released from reset
9 LPUART1	LPUART1 0b - Peripheral is held in reset 1b - Peripheral is released from reset
8 LPUART0	LPUART0 0b - Peripheral is held in reset 1b - Peripheral is released from reset
7-5 —	reserved reserved
4 LPSPI0	LPSPI0 0b - Peripheral is held in reset 1b - Peripheral is released from reset
3-1 —	reserved reserved
0 LPI2C0	LPI2C0 0b - Peripheral is held in reset 1b - Peripheral is released from reset

All information provided in this document is subject to legal disclaimers.

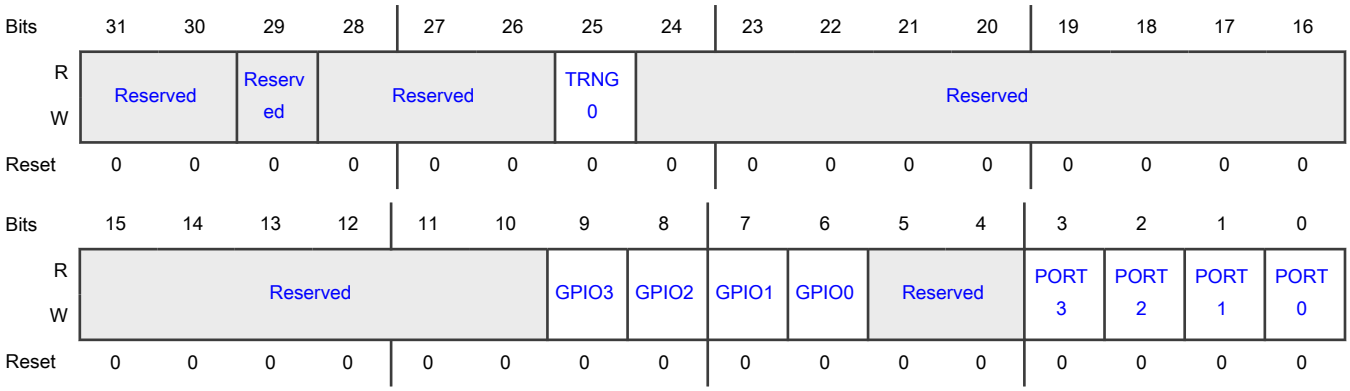
© 2026 NXP B.V. All rights reserved.

13.4.1.6 Peripheral Reset Control 2 (MRCC_GLB_RST2)

Offset

Register	Offset
MRCC_GLB_RST2	20h

Diagram



Fields

Field	Function
31-30 —	reserved reserved
29 —	reserved reserved
28-26 —	reserved reserved
25 TRNG0	TRNG0 0b - Peripheral is held in reset 1b - Peripheral is released from reset
24-10 —	reserved reserved
9 GPIO3	GPIO3 0b - Peripheral is held in reset 1b - Peripheral is released from reset
8	GPIO2

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
GPIO2	0b - Peripheral is held in reset 1b - Peripheral is released from reset
7 GPIO1	GPIO1 0b - Peripheral is held in reset 1b - Peripheral is released from reset
6 GPIO0	GPIO0 0b - Peripheral is held in reset 1b - Peripheral is released from reset
5-4 —	reserved reserved
3 PORT3	PORT3 0b - Peripheral is held in reset 1b - Peripheral is released from reset
2 PORT2	PORT2 0b - Peripheral is held in reset 1b - Peripheral is released from reset
1 PORT1	PORT1 0b - Peripheral is held in reset 1b - Peripheral is released from reset
0 PORT0	PORT0 0b - Peripheral is held in reset 1b - Peripheral is released from reset

13.4.1.7 AHB Clock Control 0 (MRCC_GLB_CC0)

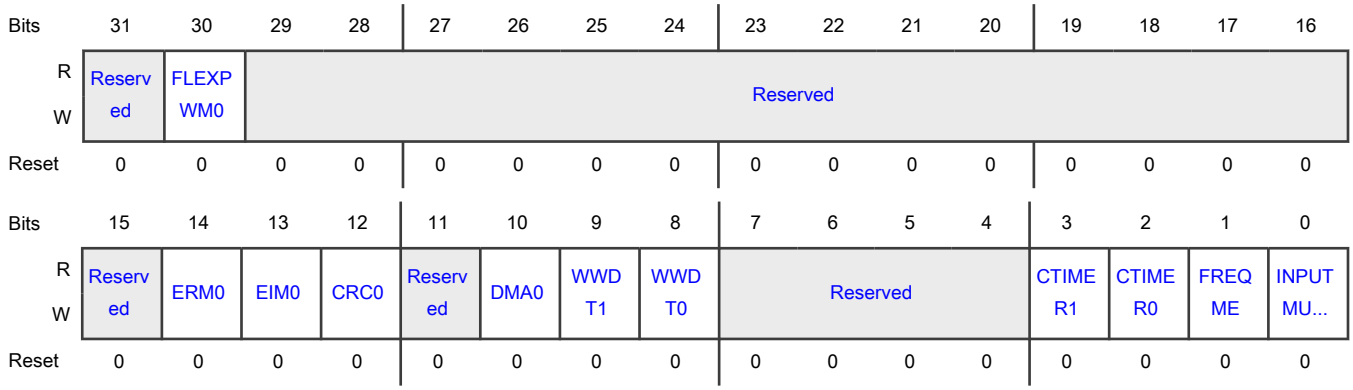
Offset

Register	Offset
MRCC_GLB_CC0	60h

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram



Fields

Field	Function
31 —	reserved reserved
30 FLEXPWM0	FLEXPWM0 0b - Peripheral clock is disabled 1b - Peripheral clock is enabled
29-15 —	reserved reserved
14 ERM0	ERM0 0b - Peripheral clock is disabled 1b - Peripheral clock is enabled
13 EIM0	EIM0 0b - Peripheral clock is disabled 1b - Peripheral clock is enabled
12 CRC0	CRC0 0b - Peripheral clock is disabled 1b - Peripheral clock is enabled
11 —	reserved reserved
10 DMA0	DMA0 0b - Peripheral clock is disabled 1b - Peripheral clock is enabled
9	WWDT1

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
WWDT1	0b - Peripheral clock is disabled 1b - Peripheral clock is enabled
8 WWDT0	WWDT0 0b - Peripheral clock is disabled 1b - Peripheral clock is enabled
7-4 —	reserved reserved
3 CTIMER1	CTIMER1 0b - Peripheral clock is disabled 1b - Peripheral clock is enabled
2 CTIMER0	CTIMER0 0b - Peripheral clock is disabled 1b - Peripheral clock is enabled
1 FREQME	FREQME 0b - Peripheral clock is disabled 1b - Peripheral clock is enabled
0 INPUTMUX0	INPUTMUX0 0b - Peripheral clock is disabled 1b - Peripheral clock is enabled

13.4.1.8 AHB Clock Control Set n (MRCC_GLB_CC0_SET - MRCC_GLB_CC2_SET)

Offset

Register	Offset
MRCC_GLB_CC0_SET	64h
MRCC_GLB_CC1_SET	74h
MRCC_GLB_CC2_SET	84h

Function

Writing a 1 to a bit position in a write-only MRCC_GLB_CCn_SET register sets the corresponding position in MRCC_GLB_CCn.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R																
W	DATA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R																
W	DATA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-0 DATA	Data array value, refer to corresponding position in MRCC_GLB_CCn.

13.4.1.9 AHB Clock Control Clear n (MRCC_GLB_CC0_CLR - MRCC_GLB_CC2_CLR)

Offset

Register	Offset
MRCC_GLB_CC0_CLR	68h
MRCC_GLB_CC1_CLR	78h
MRCC_GLB_CC2_CLR	88h

Function

Writing a 1 to a bit position in a write-only MRCC_GLB_CCn_CLR register clears the corresponding position in MRCC_GLB_CCn.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R																
W	DATA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R																
W	DATA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Fields

Field	Function
31-0 DATA	Data array value, refer to corresponding position in MRCC_GLB_CCn.

13.4.1.10 AHB Clock Control 1 (MRCC_GLB_CC1)

Offset

Register	Offset
MRCC_GLB_CC1	70h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved						OPAMP0		Reserved			CMP0	Reserved			ADC0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved				LPUART3	LPUART2	LPUART1	LPUART0	Reserved			LPSPI0	Reserved			LPI2C0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-26 —	reserved reserved
25 OPAMP0	OPAMP0 0b - Peripheral clock is disabled 1b - Peripheral clock is enabled
24-21 —	reserved reserved
20 CMP0	CMP0 0b - Peripheral clock is disabled 1b - Peripheral clock is enabled

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
19-17 —	reserved reserved
16 ADC0	ADC0 0b - Peripheral clock is disabled 1b - Peripheral clock is enabled
15-12 —	reserved reserved
11 LPUART3	LPUART3 0b - Peripheral clock is disabled 1b - Peripheral clock is enabled
10 LPUART2	LPUART2 0b - Peripheral clock is disabled 1b - Peripheral clock is enabled
9 LPUART1	LPUART1 0b - Peripheral clock is disabled 1b - Peripheral clock is enabled
8 LPUART0	LPUART0 0b - Peripheral clock is disabled 1b - Peripheral clock is enabled
7-5 —	reserved reserved
4 LPSPI0	LPSPI0 0b - Peripheral clock is disabled 1b - Peripheral clock is enabled
3-1 —	reserved reserved
0 LPI2C0	LPI2C0 0b - Peripheral clock is disabled 1b - Peripheral clock is enabled

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

13.4.1.11 AHB Clock Control 2 (MRCC_GLB_CC2)

Offset

Register	Offset
MRCC_GLB_CC2	80h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved			Reserved		SECCON	TRNG0	Reserved								
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved			RAMA	Reserved		GPIO3	GPIO2	GPIO1	GPIO0	Reserved		PORT3	PORT2	PORT1	PORT0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-29 —	reserved reserved
28-27 —	reserved reserved
26 SECCON	SECCON 0b - Peripheral clock is disabled 1b - Peripheral clock is enabled
25 TRNG0	TRNG0 0b - Peripheral clock is disabled 1b - Peripheral clock is enabled
24-13 —	reserved reserved
12 RAMA	RAMA 0b - Peripheral clock is disabled 1b - Peripheral clock is enabled
11-10	reserved

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
—	reserved
9 GPIO3	GPIO3 0b - Peripheral clock is disabled 1b - Peripheral clock is enabled
8 GPIO2	GPIO2 0b - Peripheral clock is disabled 1b - Peripheral clock is enabled
7 GPIO1	GPIO1 0b - Peripheral clock is disabled 1b - Peripheral clock is enabled
6 GPIO0	GPIO0 0b - Peripheral clock is disabled 1b - Peripheral clock is enabled
5-4 —	reserved reserved
3 PORT3	PORT3 0b - Peripheral clock is disabled 1b - Peripheral clock is enabled
2 PORT2	PORT2 0b - Peripheral clock is disabled 1b - Peripheral clock is enabled
1 PORT1	PORT1 0b - Peripheral clock is disabled 1b - Peripheral clock is enabled
0 PORT0	PORT0 0b - Peripheral clock is disabled 1b - Peripheral clock is enabled

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

13.4.1.12 Control Automatic Clock Gating 0 (MRCC_GLB_ACC0)

Offset

Register	Offset
MRCC_GLB_ACC0	C0h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved	FLEXPWM0	Reserved													
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved	ERM0	EIM0	CRC0	Reserved	DMA0	WWDT1	WWDT0	Reserved				CTIMER1	CTIMER0	FREQME	INPUTMU...
W																
Reset	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0

Fields

Field	Function
31 —	reserved reserved
30 FLEXPWM0	FLEXPWM0 0b - Automatic clock gating is disabled 1b - Automatic clock gating is enabled
29-15 —	reserved reserved
14 ERM0	ERM0 0b - Automatic clock gating is disabled 1b - Automatic clock gating is enabled
13 EIM0	EIM0 0b - Automatic clock gating is disabled 1b - Automatic clock gating is enabled
12 CRC0	CRC0 0b - Automatic clock gating is disabled 1b - Automatic clock gating is enabled

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
11 —	reserved reserved
10 DMA0	DMA0 0b - Automatic clock gating is disabled 1b - Automatic clock gating is enabled
9 WWDT1	WWDT1 0b - Automatic clock gating is disabled 1b - Automatic clock gating is enabled
8 WWDT0	WWDT0 0b - Automatic clock gating is disabled 1b - Automatic clock gating is enabled
7-4 —	reserved reserved
3 CTIMER1	CTIMER1 0b - Automatic clock gating is disabled 1b - Automatic clock gating is enabled
2 CTIMER0	CTIMER0 0b - Automatic clock gating is disabled 1b - Automatic clock gating is enabled
1 FREQME	FREQME 0b - Automatic clock gating is disabled 1b - Automatic clock gating is enabled
0 INPUTMUX0	INPUTMUX0 0b - Automatic clock gating is disabled 1b - Automatic clock gating is enabled

13.4.1.13 Control Automatic Clock Gating 1 (MRCC_GLB_ACC1)

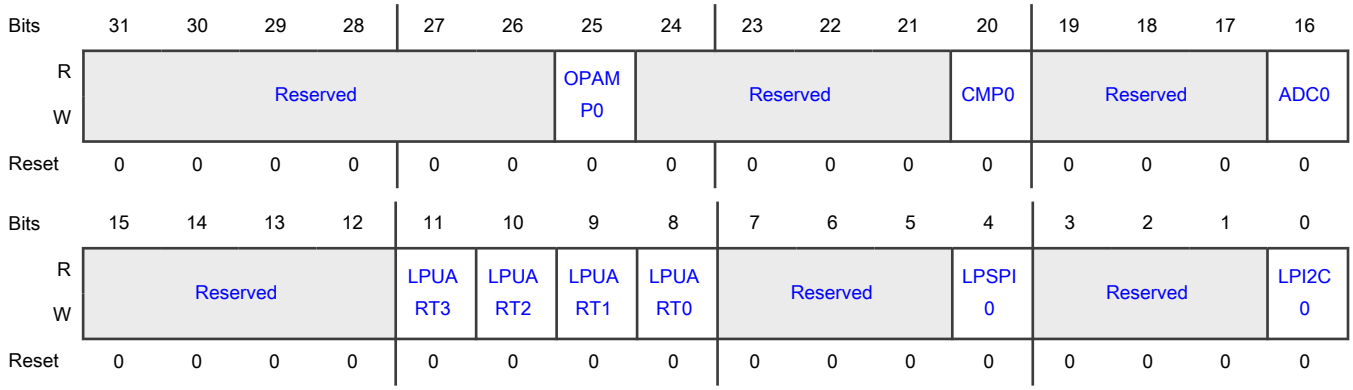
Offset

Register	Offset
MRCC_GLB_ACC1	C4h

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram



Fields

Field	Function
31-26 —	reserved reserved
25 OPAMP0	OPAMP0 0b - Automatic clock gating is disabled 1b - Automatic clock gating is enabled
24-21 —	reserved reserved
20 CMP0	CMP0 0b - Automatic clock gating is disabled 1b - Automatic clock gating is enabled
19-17 —	reserved reserved
16 ADC0	ADC0 0b - Automatic clock gating is disabled 1b - Automatic clock gating is enabled
15-12 —	reserved reserved
11 LPUART3	LPUART3 0b - Automatic clock gating is disabled 1b - Automatic clock gating is enabled
10	LPUART2

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
LPUART2	0b - Automatic clock gating is disabled 1b - Automatic clock gating is enabled
9 LPUART1	LPUART1 0b - Automatic clock gating is disabled 1b - Automatic clock gating is enabled
8 LPUART0	LPUART0 0b - Automatic clock gating is disabled 1b - Automatic clock gating is enabled
7-5 —	reserved reserved
4 LPSPI0	LPSPI0 0b - Automatic clock gating is disabled 1b - Automatic clock gating is enabled
3-1 —	reserved reserved
0 LPI2C0	LPI2C0 0b - Automatic clock gating is disabled 1b - Automatic clock gating is enabled

13.4.1.14 Control Automatic Clock Gating 2 (MRCC_GLB_ACC2)

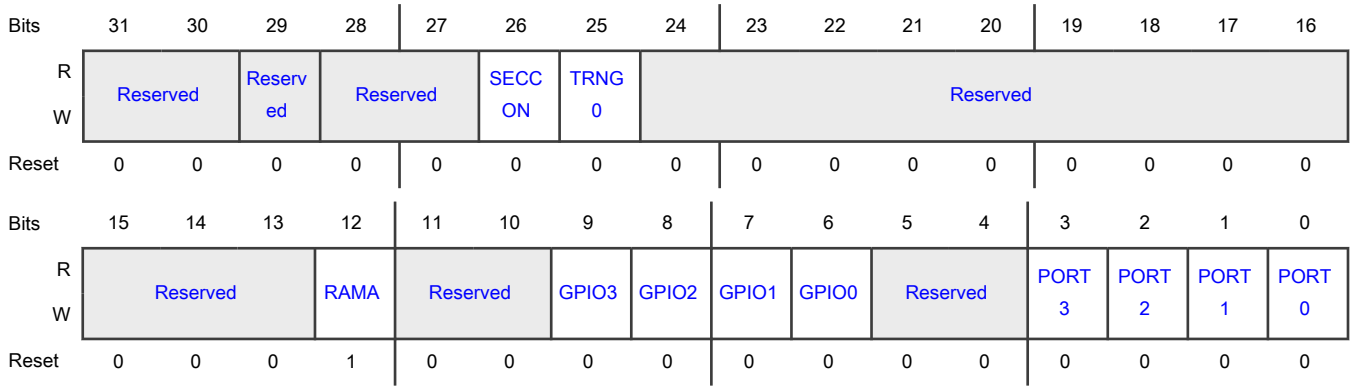
Offset

Register	Offset
MRCC_GLB_ACC2	C8h

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram



Fields

Field	Function
31-30 —	reserved reserved
29 —	reserved reserved
28-27 —	reserved reserved
26 SECCON	SECCON 0b - Automatic clock gating is disabled 1b - Automatic clock gating is enabled
25 TRNG0	TRNG0 0b - Automatic clock gating is disabled 1b - Automatic clock gating is enabled
24-13 —	reserved reserved
12 RAMA	RAMA 0b - Automatic clock gating is disabled 1b - Automatic clock gating is enabled
11-10 —	reserved reserved
9 GPIO3	GPIO3 0b - Automatic clock gating is disabled

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	1b - Automatic clock gating is enabled
8 GPIO2	GPIO2 0b - Automatic clock gating is disabled 1b - Automatic clock gating is enabled
7 GPIO1	GPIO1 0b - Automatic clock gating is disabled 1b - Automatic clock gating is enabled
6 GPIO0	GPIO0 0b - Automatic clock gating is disabled 1b - Automatic clock gating is enabled
5-4 —	reserved reserved
3 PORT3	PORT3 0b - Automatic clock gating is disabled 1b - Automatic clock gating is enabled
2 PORT2	PORT2 0b - Automatic clock gating is disabled 1b - Automatic clock gating is enabled
1 PORT1	PORT1 0b - Automatic clock gating is disabled 1b - Automatic clock gating is enabled
0 PORT0	PORT0 0b - Automatic clock gating is disabled 1b - Automatic clock gating is enabled

13.4.1.15 CTIMER0 clock selection control (MRCC_CTIMER0_CLKSEL)

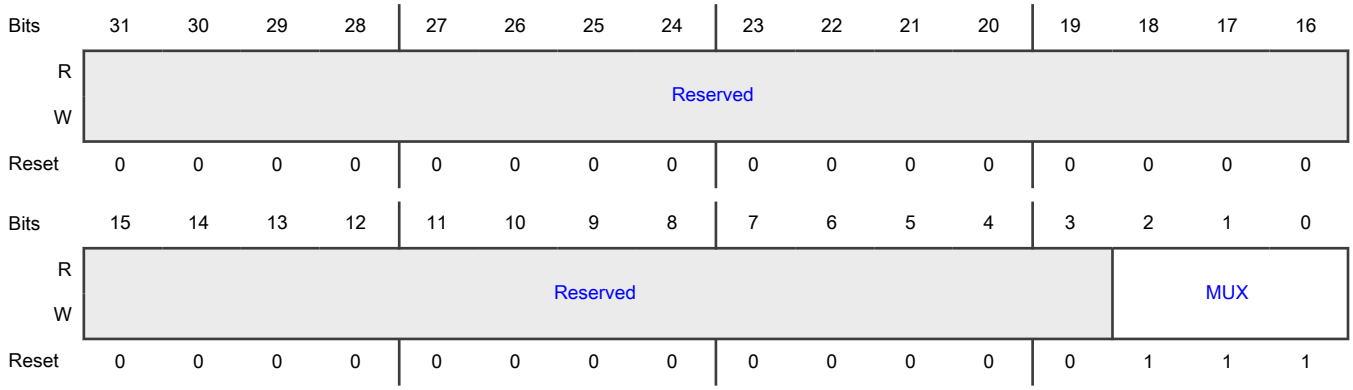
Offset

Register	Offset
MRCC_CTIMER0_CLKSEL	100h

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram



Fields

Field	Function
31-3 —	reserved reserved
2-0 MUX	Functional Clock Mux Select Before change peripheral's funcitional clock source, should configure the module's MRCC_GLB_CC and MRCC_GLB_ACC bits to 0. 000b - FRO_LF_DIV 001b - FRO_HF_GATED 011b - CLK_IN 100b - CLK_16K 101b - CLK_1M 111b - Reserved(NO Clock)

13.4.1.16 CTIMER0 clock divider control (MRCC_CTIMER0_CLKDIV)

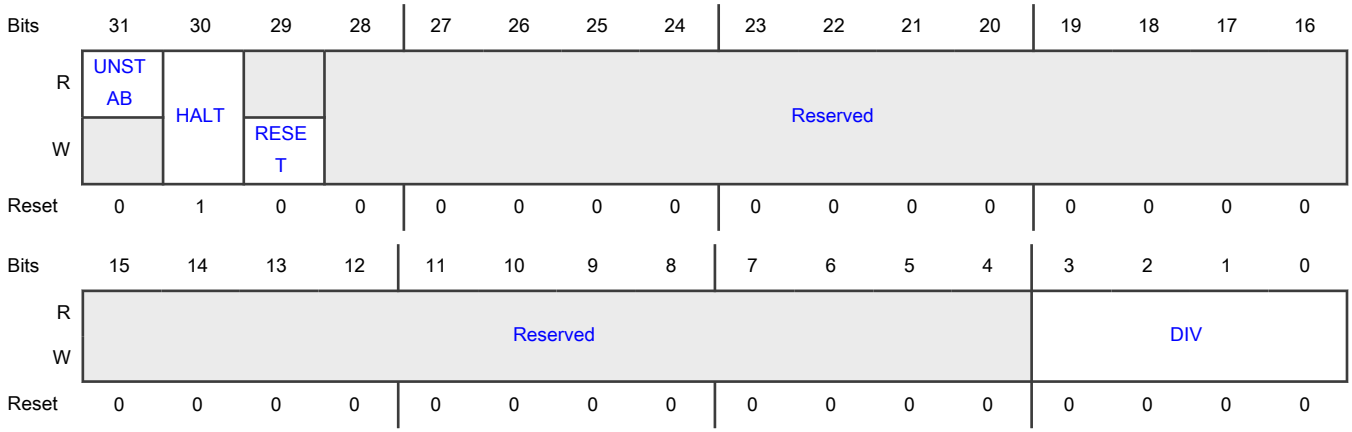
Offset

Register	Offset
MRCC_CTIMER0_CLKDIV	104h

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram



Fields

Field	Function
31 UNSTAB	Divider status flag Determines the stability of the divider clock. 0b - Divider clock is stable 1b - Clock frequency isn't stable
30 HALT	Halt divider counter Halt divider counter 0b - Divider clock is running 1b - Divider clock is stopped
29 RESET	Reset divider counter Reset divider counter 0b - Divider isn't reset 1b - Divider is reset
28-4 —	reserved reserved
3-0 DIV	Functional Clock Divider Before change peripheral's funcitional clock source, should configure the module's MRCC_GLB_CC and MRCC_GLB_ACC bits to 0. divider value = (DIV+1)

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

13.4.1.17 CTIMER1 clock selection control (MRCC_CTIMER1_CLKSEL)

Offset

Register	Offset
MRCC_CTIMER1_CLKSEL	108h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved													MUX		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1

Fields

Field	Function
31-3 —	reserved reserved
2-0 MUX	Functional Clock Mux Select Before change peripheral's funcitional clock source, should configure the module's MRCC_GLB_CC and MRCC_GLB_ACC bits to 0. 000b - FRO_LF_DIV 001b - FRO_HF_GATED 011b - CLK_IN 100b - CLK_16K 101b - CLK_1M 111b - Reserved(NO Clock)

All information provided in this document is subject to legal disclaimers.

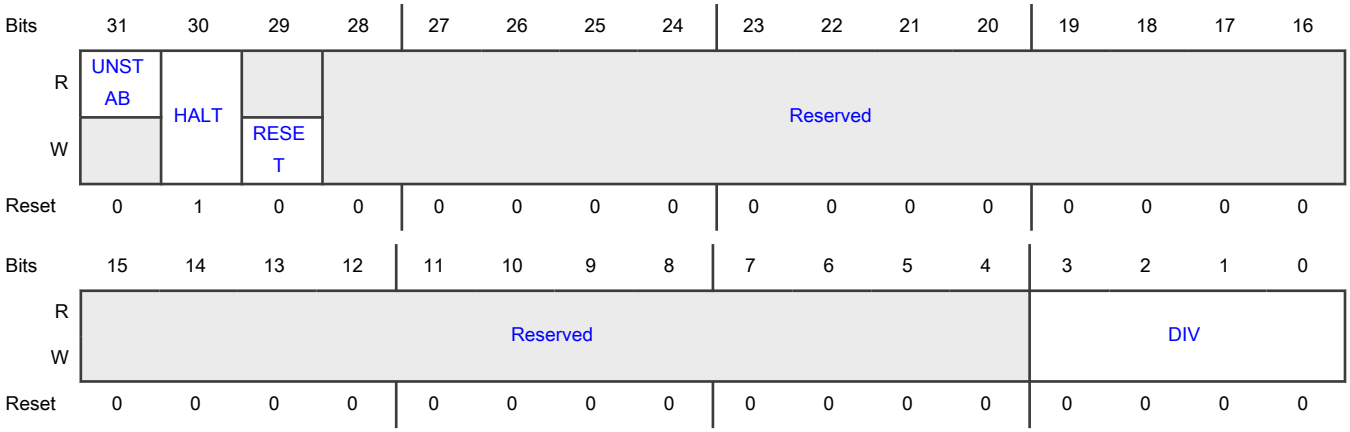
© 2026 NXP B.V. All rights reserved.

13.4.1.18 CTIMER1 clock divider control (MRCC_CTIMER1_CLKDIV)

Offset

Register	Offset
MRCC_CTIMER1_CLKDIV	10Ch

Diagram



Fields

Field	Function
31 UNSTAB	Divider status flag Determines the stability of the divider clock. 0b - Divider clock is stable 1b - Clock frequency isn't stable
30 HALT	Halt divider counter Halt divider counter 0b - Divider clock is running 1b - Divider clock is stopped
29 RESET	Reset divider counter Reset divider counter 0b - Divider isn't reset 1b - Divider is reset
28-4 —	reserved reserved

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

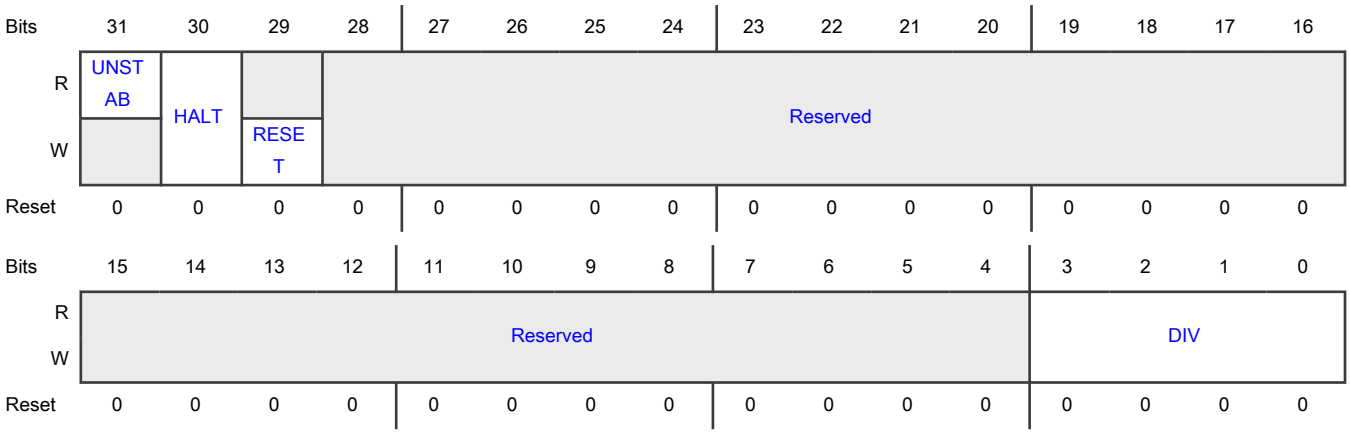
Field	Function
3-0 DIV	Functional Clock Divider Before change peripheral's funcitional clock source, should configure the module's MRCC_GLB_CC and MRCC_GLB_ACC bits to 0. divider value = (DIV+1)

13.4.1.19 WWDT0 clock divider control (MRCC_WWDT0_CLKDIV)

Offset

Register	Offset
MRCC_WWDT0_CLKDIV	114h

Diagram



Fields

Field	Function
31 UNSTAB	Divider status flag Determines the stability of the divider clock. 0b - Divider clock is stable 1b - Clock frequency isn't stable
30 HALT	Halt divider counter Halt divider counter 0b - Divider clock is running 1b - Divider clock is stopped

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

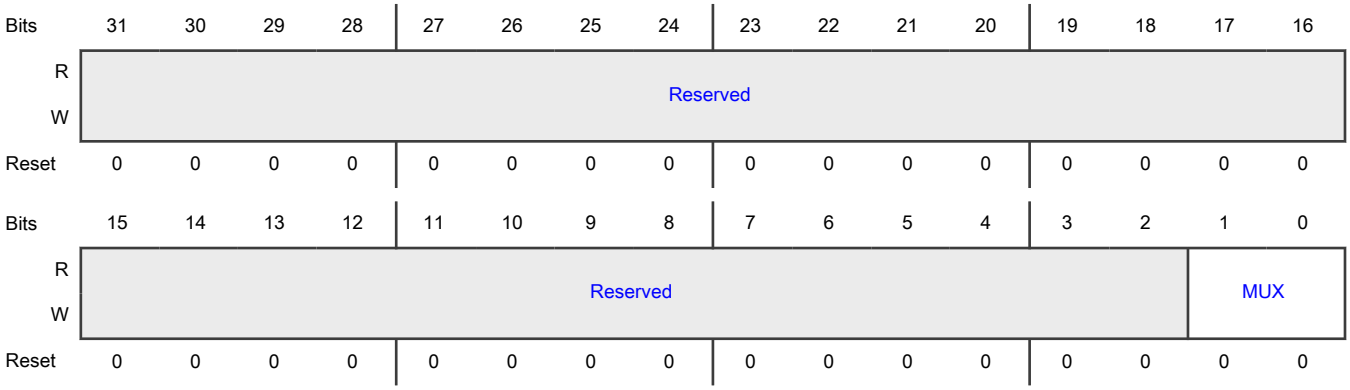
Field	Function
29 RESET	Reset divider counter Reset divider counter 0b - Divider isn't reset 1b - Divider is reset
28-4 —	reserved reserved
3-0 DIV	Functional Clock Divider Before change peripheral's funcitional clock source, should configure the module's MRCC_GLB_CC and MRCC_GLB_ACC bits to 0. divider value = (DIV+1)

13.4.1.20 WWDT1 clock selection control (MRCC_WWDT1_CLKSEL)

Offset

Register	Offset
MRCC_WWDT1_CLKSEL	118h

Diagram



Fields

Field	Function
31-2 —	reserved reserved
1-0	Functional Clock Mux Select

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

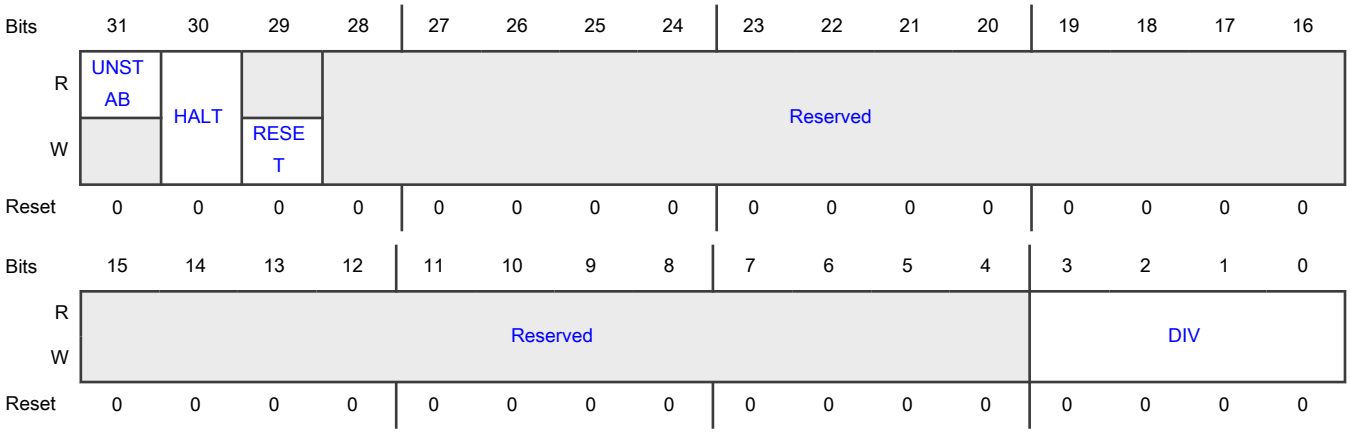
Field	Function
MUX	Before change peripheral's funcitional clock source, should configure the module's MRCC_GLB_CC and MRCC_GLB_ACC bits to 0. 00b - CLK_16K 01b - FRO_HF_DIV 10b - CLK_1M 11b - CLK_1M

13.4.1.21 WWDT1 clock divider control (MRCC_WWDT1_CLKDIV)

Offset

Register	Offset
MRCC_WWDT1_CLKDIV	11Ch

Diagram



Fields

Field	Function
31 UNSTAB	Divider status flag Determines the stability of the divider clock. 0b - Divider clock is stable 1b - Clock frequency isn't stable
30 HALT	Halt divider counter Halt divider counter

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

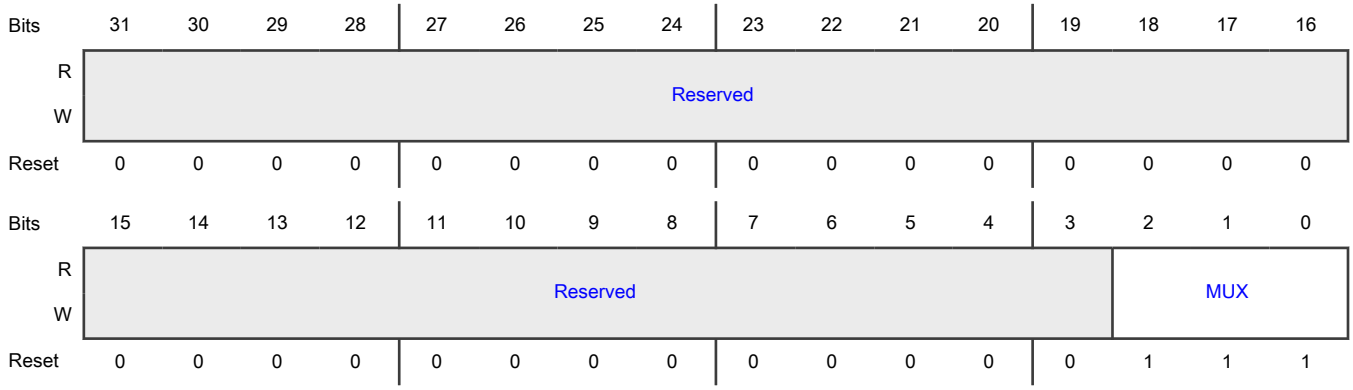
Field	Function
	0b - Divider clock is running 1b - Divider clock is stopped
29 RESET	Reset divider counter Reset divider counter 0b - Divider isn't reset 1b - Divider is reset
28-4 —	reserved reserved
3-0 DIV	Functional Clock Divider Before change peripheral's funcitional clock source, should configure the module's MRCC_GLB_CC and MRCC_GLB_ACC bits to 0. divider value = (DIV+1)

13.4.1.22 LPI2C0 clock selection control (MRCC_LPI2C0_CLKSEL)

Offset

Register	Offset
MRCC_LPI2C0_CLKSEL	120h

Diagram



Fields

Field	Function
31-3 —	reserved reserved

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

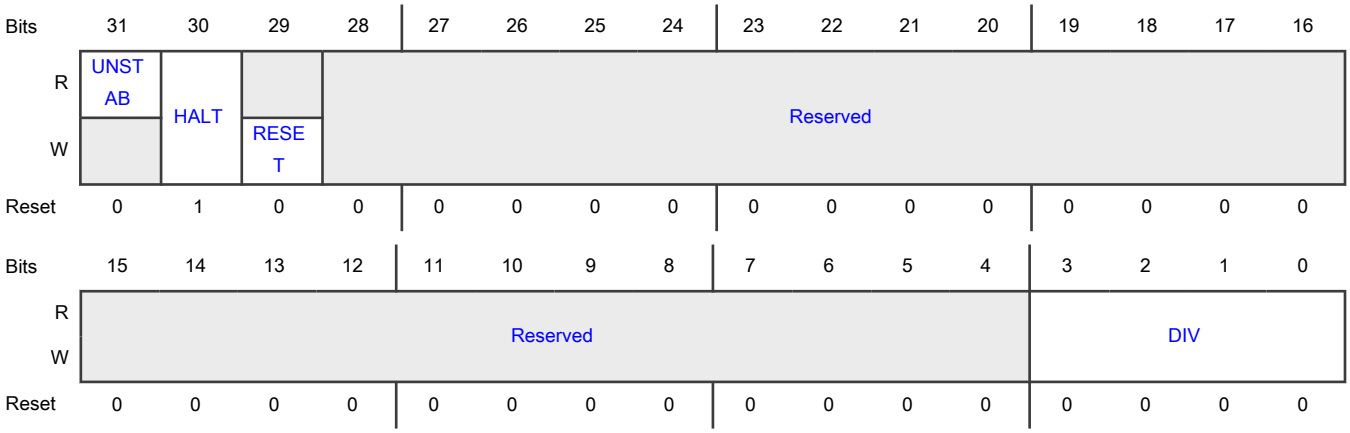
Field	Function
2-0 MUX	Functional Clock Mux Select Before change peripheral's funcitional clock source, should configure the module's MRCC_GLB_CC and MRCC_GLB_ACC bits to 0. 000b - FRO_LF_DIV 010b - FRO_HF_DIV 011b - CLK_IN 101b - CLK_1M 111b - Reserved(NO Clock)

13.4.1.23 LPI2C0 clock divider control (MRCC_LPI2C0_CLKDIV)

Offset

Register	Offset
MRCC_LPI2C0_CLKDIV	124h

Diagram



Fields

Field	Function
31 UNSTAB	Divider status flag Determines the stability of the divider clock. 0b - Divider clock is stable 1b - Clock frequency isn't stable

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

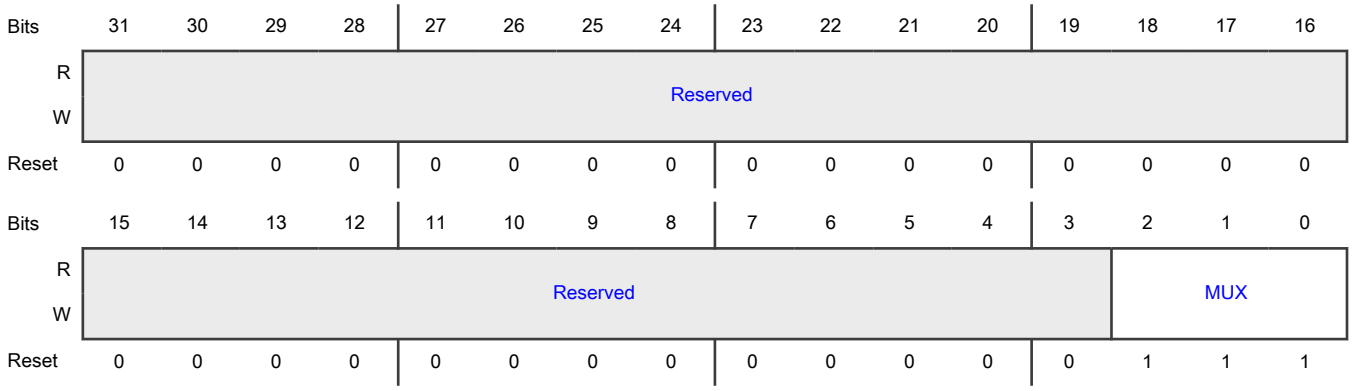
Field	Function
30 HALT	Halt divider counter Halt divider counter 0b - Divider clock is running 1b - Divider clock is stopped
29 RESET	Reset divider counter Reset divider counter 0b - Divider isn't reset 1b - Divider is reset
28-4 —	reserved reserved
3-0 DIV	Functional Clock Divider Before change peripheral's funcitional clock source, should configure the module's MRCC_GLB_CC and MRCC_GLB_ACC bits to 0. divider value = (DIV+1)

13.4.1.24 LPSPI0 clock selection control (MRCC_LPSPI0_CLKSEL)

Offset

Register	Offset
MRCC_LPSPI0_CLKSEL	128h

Diagram



All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Fields

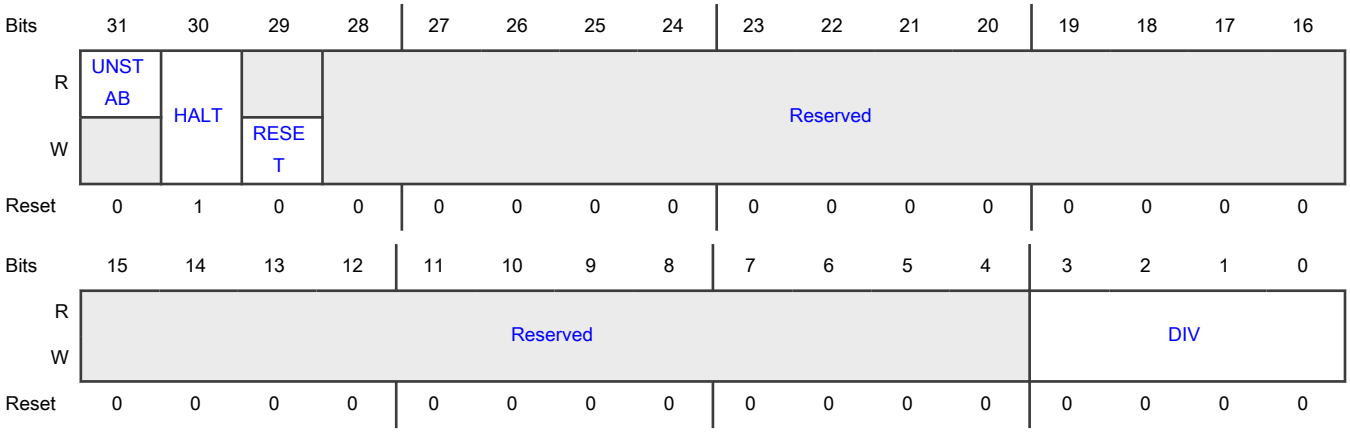
Field	Function
31-3 —	reserved reserved
2-0 MUX	Functional Clock Mux Select Before change peripheral's funcitional clock source, should configure the module's MRCC_GLB_CC and MRCC_GLB_ACC bits to 0. 000b - FRO_LF_DIV 010b - FRO_HF_DIV 011b - CLK_IN 101b - CLK_1M 111b - Reserved(NO Clock)

13.4.1.25 LPSPI0 clock divider control (MRCC_LPSPI0_CLKDIV)

Offset

Register	Offset
MRCC_LPSPI0_CLKDIV	12Ch

Diagram



Fields

Field	Function
31 UNSTAB	Divider status flag Determines the stability of the divider clock.

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

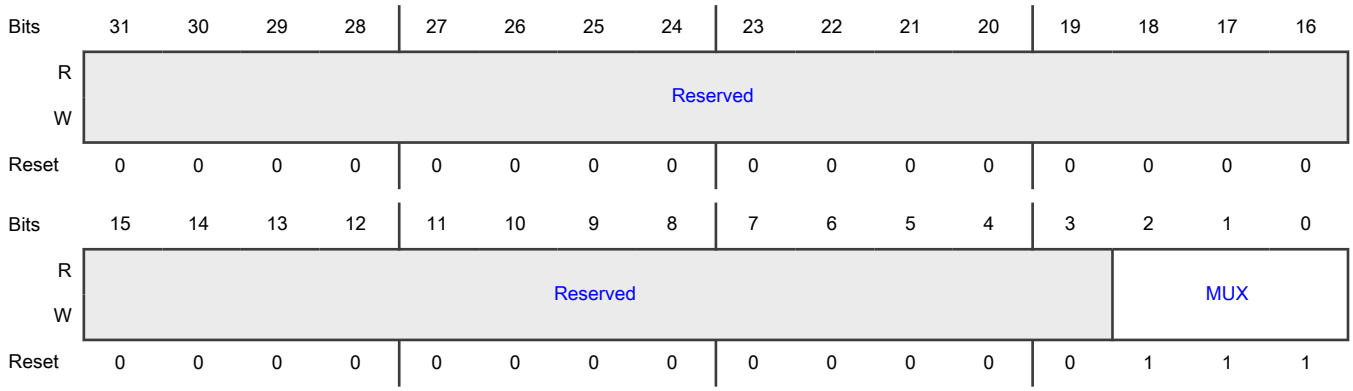
Field	Function
	0b - Divider clock is stable 1b - Clock frequency isn't stable
30 HALT	Halt divider counter Halt divider counter 0b - Divider clock is running 1b - Divider clock is stopped
29 RESET	Reset divider counter Reset divider counter 0b - Divider isn't reset 1b - Divider is reset
28-4 —	reserved reserved
3-0 DIV	Functional Clock Divider Before change peripheral's funcitional clock source, should configure the module's MRCC_GLB_CC and MRCC_GLB_ACC bits to 0. divider value = (DIV+1)

13.4.1.26 LPUART0 clock selection control (MRCC_LPUART0_CLKSEL)

Offset

Register	Offset
MRCC_LPUART0_CLKSEL	130h

Diagram



All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Fields

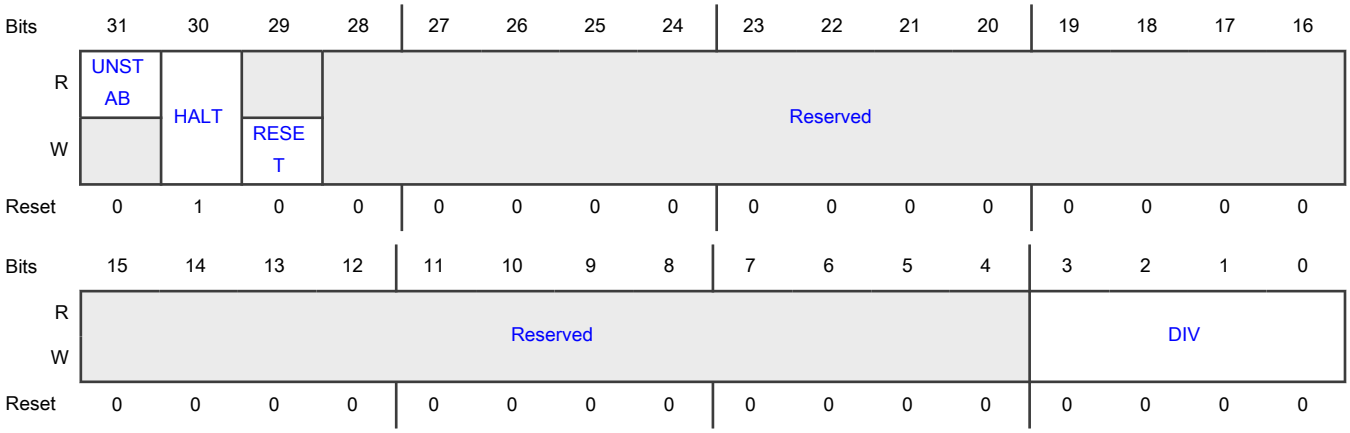
Field	Function
31-3 —	reserved reserved
2-0 MUX	Functional Clock Mux Select Before change peripheral's funcitional clock source, should configure the module's MRCC_GLB_CC and MRCC_GLB_ACC bits to 0. 000b - FRO_LF_DIV 010b - FRO_HF_DIV 011b - CLK_IN 100b - CLK_32K 101b - CLK_1M 111b - Reserved(NO Clock)

13.4.1.27 LPUART0 clock divider control (MRCC_LPUART0_CLKDIV)

Offset

Register	Offset
MRCC_LPUART0_CLKDIV	134h

Diagram



Fields

Field	Function
31	Divider status flag

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

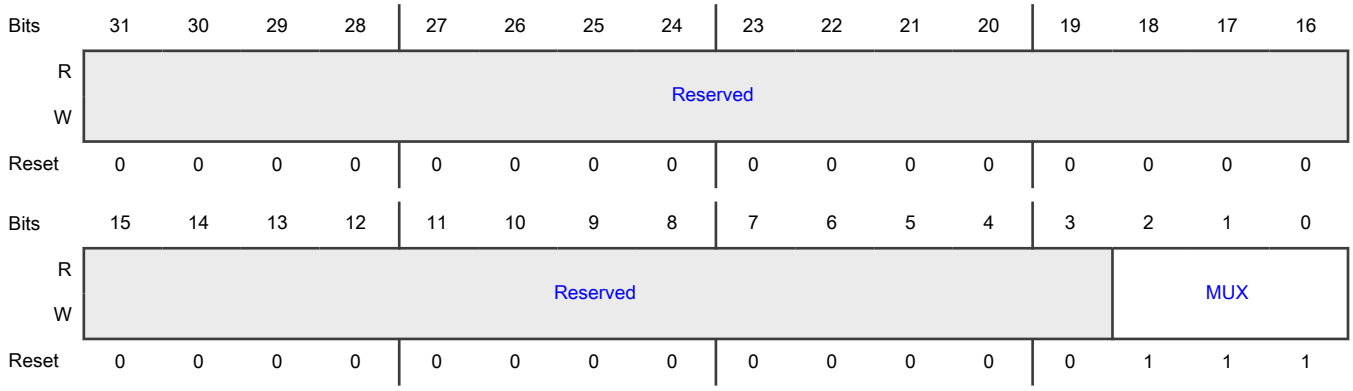
Field	Function
UNSTAB	Determines the stability of the divider clock. 0b - Divider clock is stable 1b - Clock frequency isn't stable
30 HALT	Halt divider counter Halt divider counter 0b - Divider clock is running 1b - Divider clock is stopped
29 RESET	Reset divider counter Reset divider counter 0b - Divider isn't reset 1b - Divider is reset
28-4 —	reserved reserved
3-0 DIV	Functional Clock Divider Before change peripheral's funcitional clock source, should configure the module's MRCC_GLB_CC and MRCC_GLB_ACC bits to 0. divider value = (DIV+1)

13.4.1.28 LPUART1 clock selection control (MRCC_LPUART1_CLKSEL)

Offset

Register	Offset
MRCC_LPUART1_CLKSEL	138h

Diagram



All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Fields

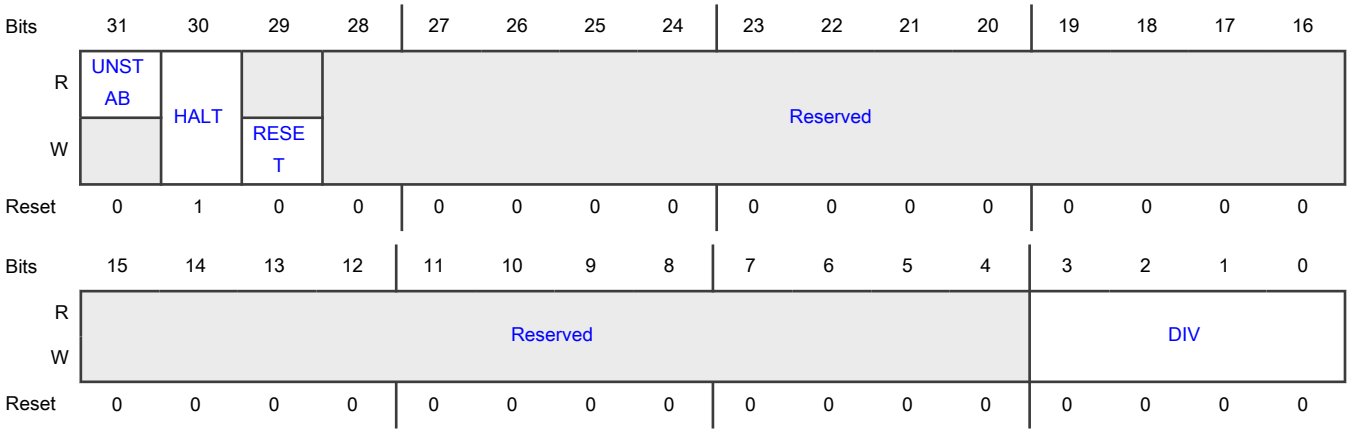
Field	Function
31-3 —	reserved reserved
2-0 MUX	Functional Clock Mux Select Before change peripheral's funcitional clock source, should configure the module's MRCC_GLB_CC and MRCC_GLB_ACC bits to 0. 000b - FRO_LF_DIV 010b - FRO_HF_DIV 011b - CLK_IN 100b - CLK_32K 101b - CLK_1M 111b - Reserved(NO Clock)

13.4.1.29 LPUART1 clock divider control (MRCC_LPUART1_CLKDIV)

Offset

Register	Offset
MRCC_LPUART1_CLKDIV	13Ch

Diagram



Fields

Field	Function
31	Divider status flag

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

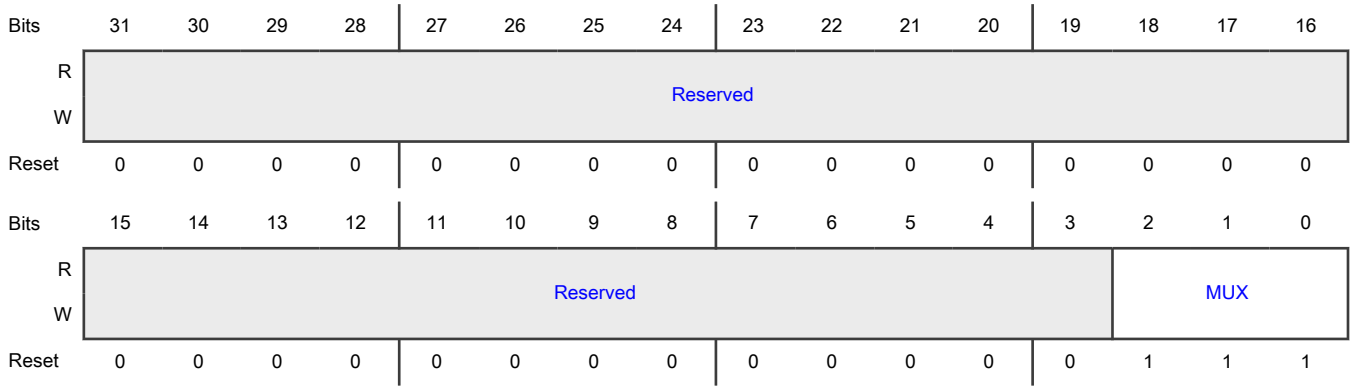
Field	Function
UNSTAB	Determines the stability of the divider clock. 0b - Divider clock is stable 1b - Clock frequency isn't stable
30 HALT	Halt divider counter Halt divider counter 0b - Divider clock is running 1b - Divider clock is stopped
29 RESET	Reset divider counter Reset divider counter 0b - Divider isn't reset 1b - Divider is reset
28-4 —	reserved reserved
3-0 DIV	Functional Clock Divider Before change peripheral's funcitional clock source, should configure the module's MRCC_GLB_CC and MRCC_GLB_ACC bits to 0. divider value = (DIV+1)

13.4.1.30 LPUART2 clock selection control (MRCC_LPUART2_CLKSEL)

Offset

Register	Offset
MRCC_LPUART2_CLKSEL	140h

Diagram



All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Fields

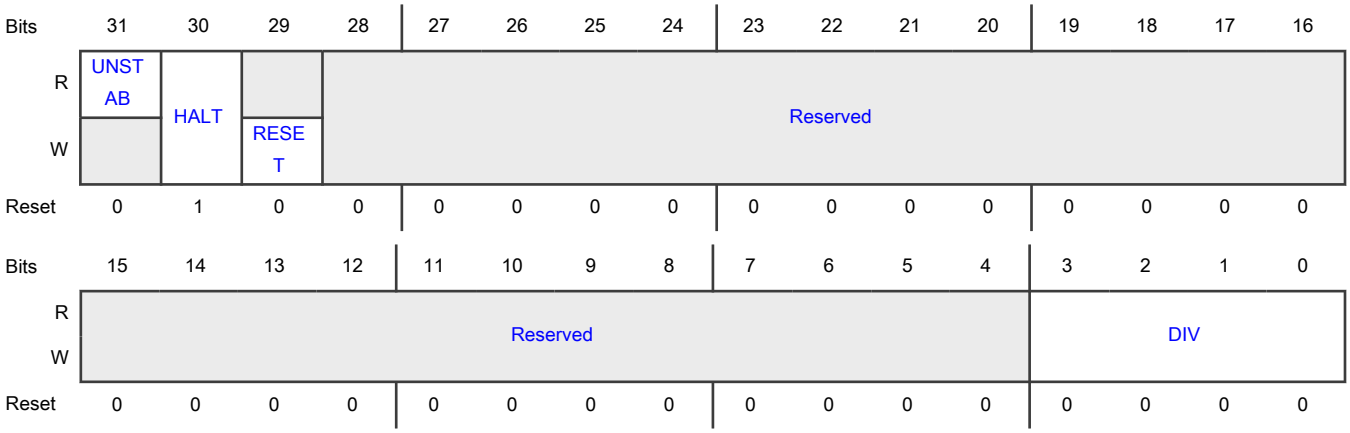
Field	Function
31-3 —	reserved reserved
2-0 MUX	Functional Clock Mux Select Before change peripheral's funcitional clock source, should configure the module's MRCC_GLB_CC and MRCC_GLB_ACC bits to 0. 000b - FRO_LF_DIV 010b - FRO_HF_DIV 011b - CLK_IN 100b - CLK_32K 101b - CLK_1M 111b - Reserved(NO Clock)

13.4.1.31 LPUART2 clock divider control (MRCC_LPUART2_CLKDIV)

Offset

Register	Offset
MRCC_LPUART2_CLKDIV	144h

Diagram



Fields

Field	Function
31	Divider status flag

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

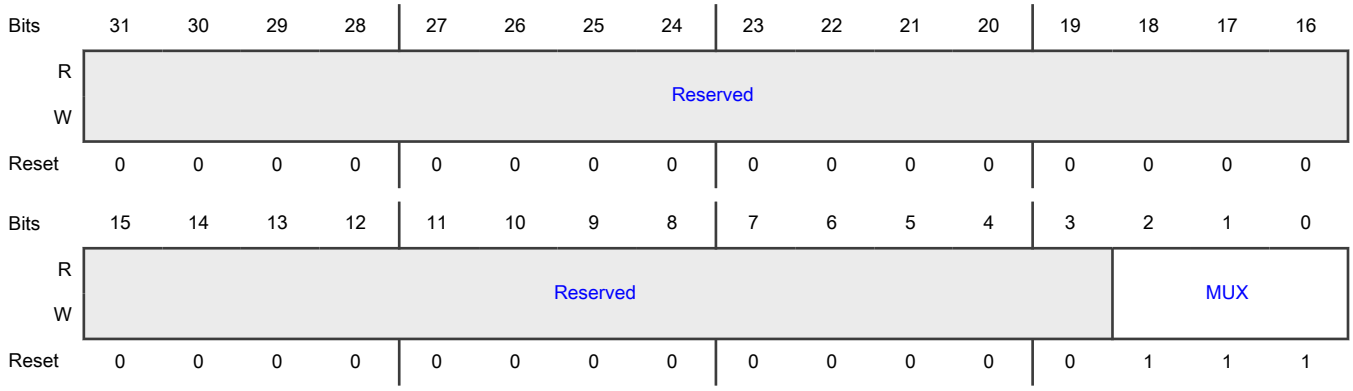
Field	Function
UNSTAB	Determines the stability of the divider clock. 0b - Divider clock is stable 1b - Clock frequency isn't stable
30 HALT	Halt divider counter Halt divider counter 0b - Divider clock is running 1b - Divider clock is stopped
29 RESET	Reset divider counter Reset divider counter 0b - Divider isn't reset 1b - Divider is reset
28-4 —	reserved reserved
3-0 DIV	Functional Clock Divider Before change peripheral's funcitional clock source, should configure the module's MRCC_GLB_CC and MRCC_GLB_ACC bits to 0. divider value = (DIV+1)

13.4.1.32 LPUART3 clock selection control (MRCC_LPUART3_CLKSEL)

Offset

Register	Offset
MRCC_LPUART3_CLKSEL	148h

Diagram



All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Fields

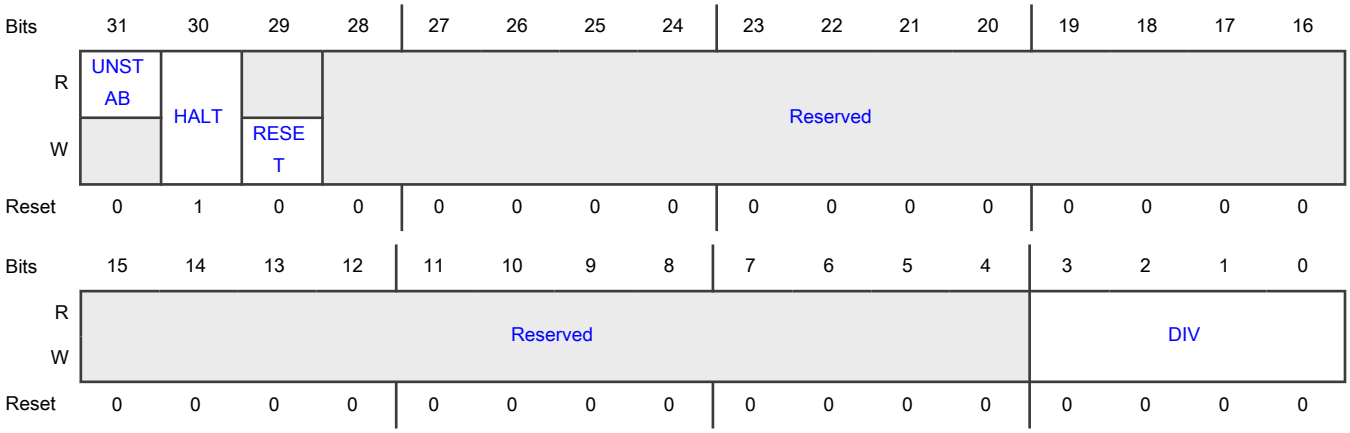
Field	Function
31-3 —	reserved reserved
2-0 MUX	Functional Clock Mux Select Before change peripheral's funcitional clock source, should configure the module's MRCC_GLB_CC and MRCC_GLB_ACC bits to 0. 000b - FRO_LF_DIV 010b - FRO_HF_DIV 011b - CLK_IN 100b - CLK_32K 101b - CLK_1M 111b - Reserved(NO Clock)

13.4.1.33 LPUART3 clock divider control (MRCC_LPUART3_CLKDIV)

Offset

Register	Offset
MRCC_LPUART3_CLKDIV	14Ch

Diagram



Fields

Field	Function
31	Divider status flag

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

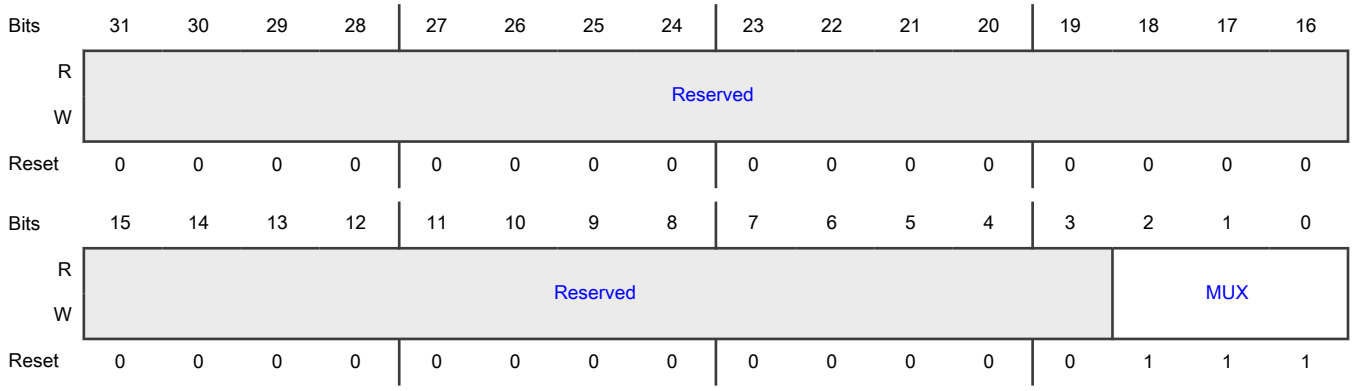
Field	Function
UNSTAB	Determines the stability of the divider clock. 0b - Divider clock is stable 1b - Clock frequency isn't stable
30 HALT	Halt divider counter Halt divider counter 0b - Divider clock is running 1b - Divider clock is stopped
29 RESET	Reset divider counter Reset divider counter 0b - Divider isn't reset 1b - Divider is reset
28-4 —	reserved reserved
3-0 DIV	Functional Clock Divider Before change peripheral's funcitional clock source, should configure the module's MRCC_GLB_CC and MRCC_GLB_ACC bits to 0. divider value = (DIV+1)

13.4.1.34 LPTMR0 clock selection control (MRCC_LPTMR0_CLKSEL)

Offset

Register	Offset
MRCC_LPTMR0_CLKSEL	150h

Diagram



All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Fields

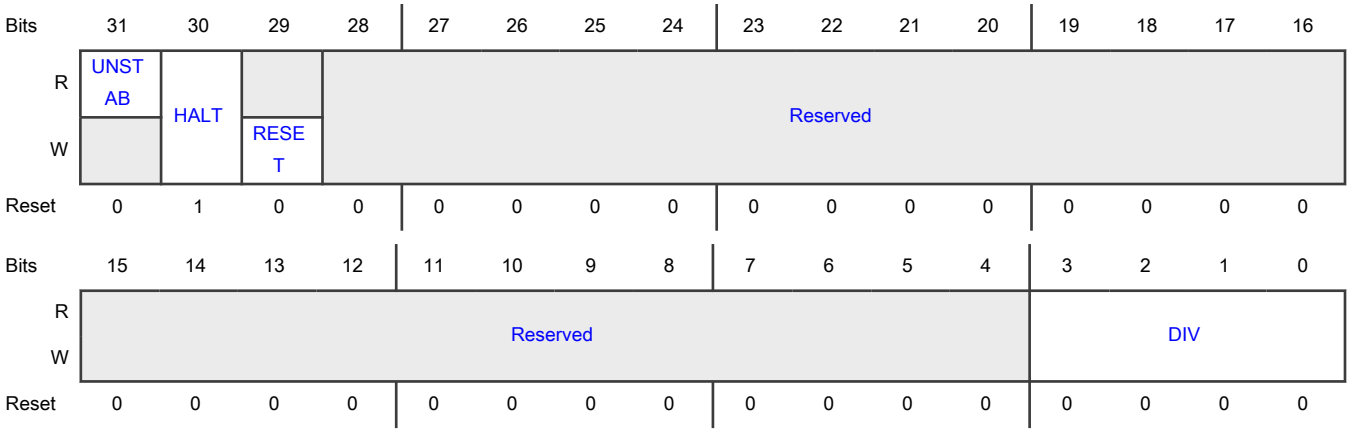
Field	Function
31-3 —	reserved reserved
2-0 MUX	Functional Clock Mux Select Before change peripheral's funcitional clock source, should configure the module's MRCC_GLB_CC and MRCC_GLB_ACC bits to 0. 000b - FRO_LF_DIV 010b - FRO_HF_DIV 011b - CLK_IN 101b - CLK_1M 111b - Reserved(NO Clock)

13.4.1.35 LPTMR0 clock divider control (MRCC_LPTMR0_CLKDIV)

Offset

Register	Offset
MRCC_LPTMR0_CLKDIV	154h

Diagram



Fields

Field	Function
31 UNSTAB	Divider status flag Determines the stability of the divider clock.

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

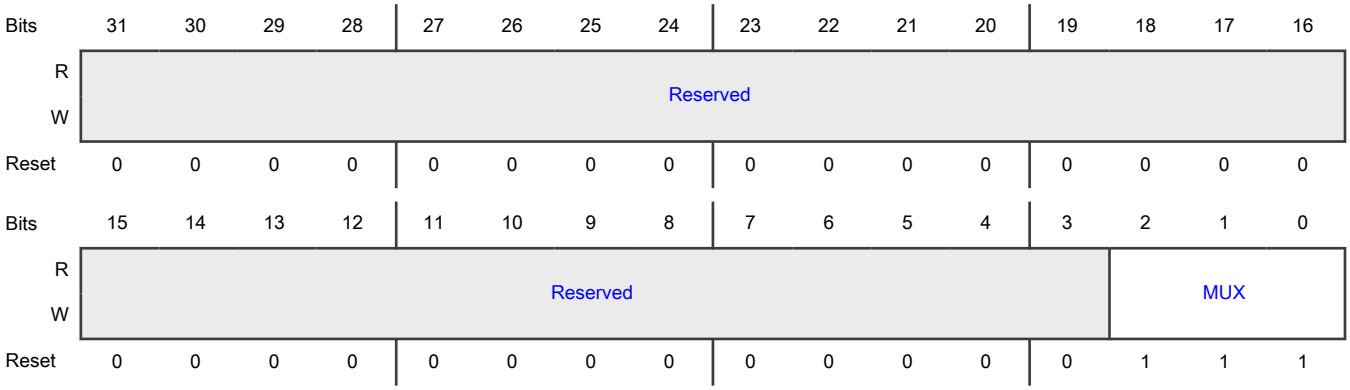
Field	Function
	0b - Divider clock is stable 1b - Clock frequency isn't stable
30 HALT	Halt divider counter Halt divider counter 0b - Divider clock is running 1b - Divider clock is stopped
29 RESET	Reset divider counter Reset divider counter 0b - Divider isn't reset 1b - Divider is reset
28-4 —	reserved reserved
3-0 DIV	Functional Clock Divider Before change peripheral's funcitional clock source, should configure the module's MRCC_GLB_CC and MRCC_GLB_ACC bits to 0. divider value = (DIV+1)

13.4.1.36 ADC clock selection control (MRCC_ADC_CLKSEL)

Offset

Register	Offset
MRCC_ADC_CLKSEL	158h

Diagram



All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Fields

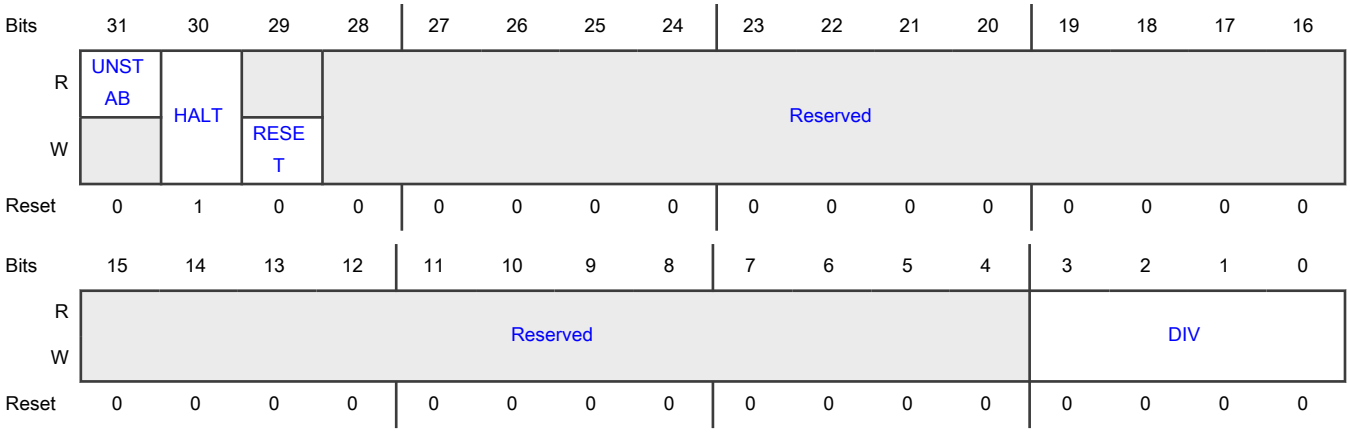
Field	Function
31-3 —	reserved reserved
2-0 MUX	Functional Clock Mux Select Before change peripheral's funcitional clock source, should configure the module's MRCC_GLB_CC and MRCC_GLB_ACC bits to 0. 000b - FRO_LF_DIV 001b - FRO_HF_GATED 011b - CLK_IN 101b - Reserved 111b - Reserved(NO Clock)

13.4.1.37 ADC clock divider control (MRCC_ADC_CLKDIV)

Offset

Register	Offset
MRCC_ADC_CLKDIV	15Ch

Diagram



Fields

Field	Function
31 UNSTAB	Divider status flag Determines the stability of the divider clock.

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

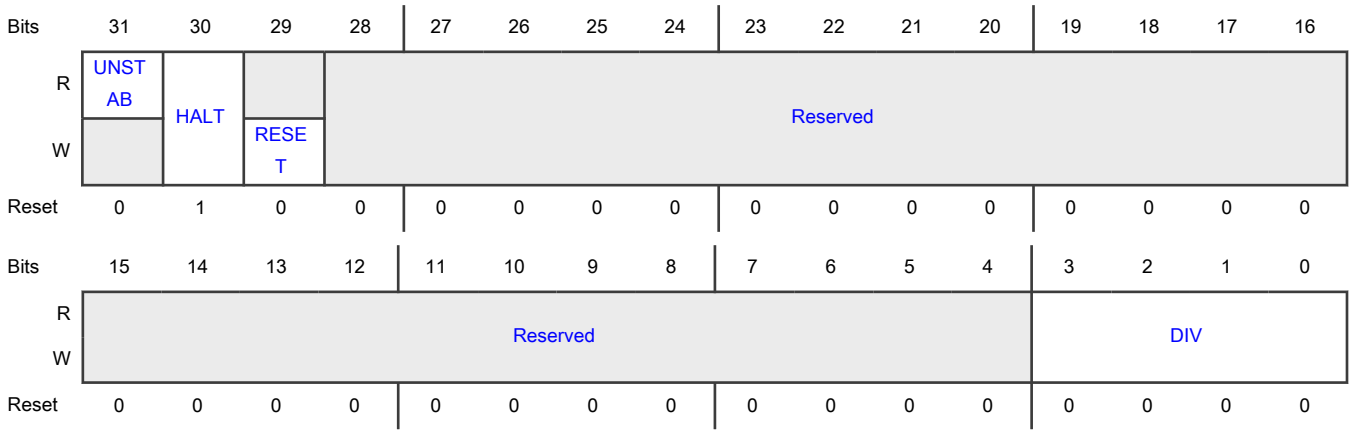
Field	Function
	0b - Divider clock is stable 1b - Clock frequency isn't stable
30 HALT	Halt divider counter Halt divider counter 0b - Divider clock is running 1b - Divider clock is stopped
29 RESET	Reset divider counter Reset divider counter 0b - Divider isn't reset 1b - Divider is reset
28-4 —	reserved reserved
3-0 DIV	Functional Clock Divider Before change peripheral's funcitional clock source, should configure the module's MRCC_GLB_CC and MRCC_GLB_ACC bits to 0. divider value = (DIV+1)

13.4.1.38 CMP0_FUNC clock divider control (MRCC_CMP0_FUNC_CLKDIV)

Offset

Register	Offset
MRCC_CMP0_FUNC_CLKDIV	164h

Diagram



All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Fields

Field	Function
31 UNSTAB	Divider status flag Determines the stability of the divider clock. 0b - Divider clock is stable 1b - Clock frequency isn't stable
30 HALT	Halt divider counter Halt divider counter 0b - Divider clock is running 1b - Divider clock is stopped
29 RESET	Reset divider counter Reset divider counter 0b - Divider isn't reset 1b - Divider is reset
28-4 —	reserved reserved
3-0 DIV	Functional Clock Divider Before change peripheral's funcitional clock source, should configure the module's MRCC_GLB_CC and MRCC_GLB_ACC bits to 0. divider value = (DIV+1)

13.4.1.39 CMP0_RR clock selection control (MRCC_CMP0_RR_CLKSEL)

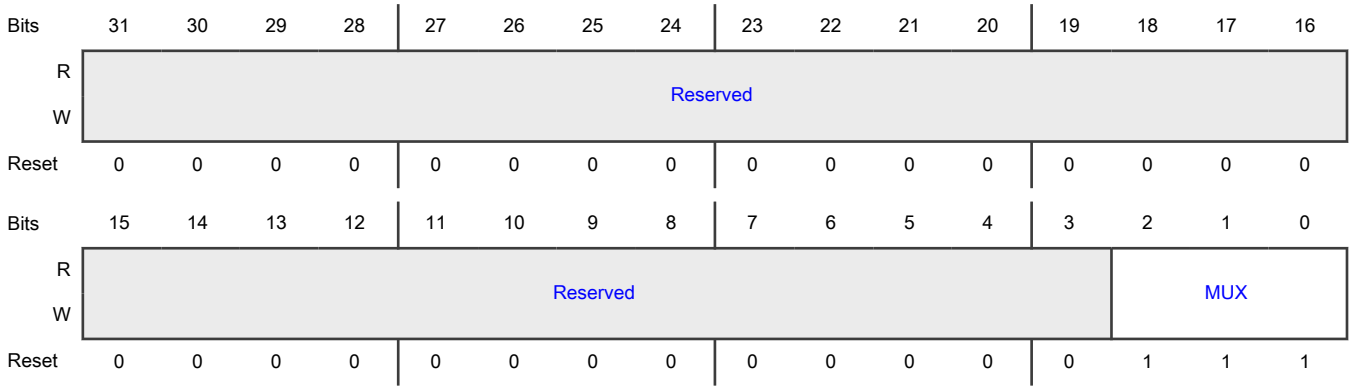
Offset

Register	Offset
MRCC_CMP0_RR_CLKSEL	168h

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram



Fields

Field	Function
31-3 —	reserved reserved
2-0 MUX	Functional Clock Mux Select Before change peripheral's funcitional clock source, should configure the module's MRCC_GLB_CC and MRCC_GLB_ACC bits to 0. 000b - FRO_LF_DIV 010b - FRO_HF_DIV 011b - CLK_IN 101b - CLK_1M 111b - Reserved(NO Clock)

13.4.1.40 CMP0_RR clock divider control (MRCC_CMP0_RR_CLKDIV)

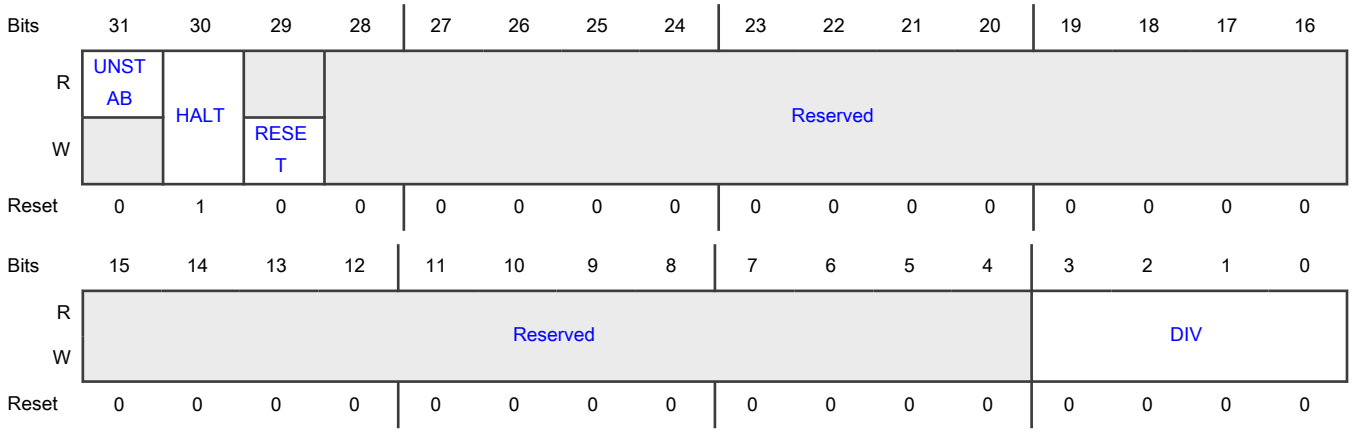
Offset

Register	Offset
MRCC_CMP0_RR_CLKDIV	16Ch

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram



Fields

Field	Function
31 UNSTAB	Divider status flag Determines the stability of the divider clock. 0b - Divider clock is stable 1b - Clock frequency isn't stable
30 HALT	Halt divider counter Halt divider counter 0b - Divider clock is running 1b - Divider clock is stopped
29 RESET	Reset divider counter Reset divider counter 0b - Divider isn't reset 1b - Divider is reset
28-4 —	reserved reserved
3-0 DIV	Functional Clock Divider Before change peripheral's funcitional clock source, should configure the module's MRCC_GLB_CC and MRCC_GLB_ACC bits to 0. divider value = (DIV+1)

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

13.4.1.41 DBG_TRACE clock selection control (MRCC_DBG_TRACE_CLKSEL)

Offset

Register	Offset
MRCC_DBG_TRACE_CLKSEL	170h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved														MUX	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-2 —	reserved reserved
1-0 MUX	Functional Clock Mux Select Before change peripheral's funcitional clock source, should configure the module's MRCC_GLB_CC and MRCC_GLB_ACC bits to 0. 00b - CPU_CLK 01b - CLK_1M 10b - CLK_16K 11b - Reserved(NO Clock)

13.4.1.42 DBG_TRACE clock divider control (MRCC_DBG_TRACE_CLKDIV)

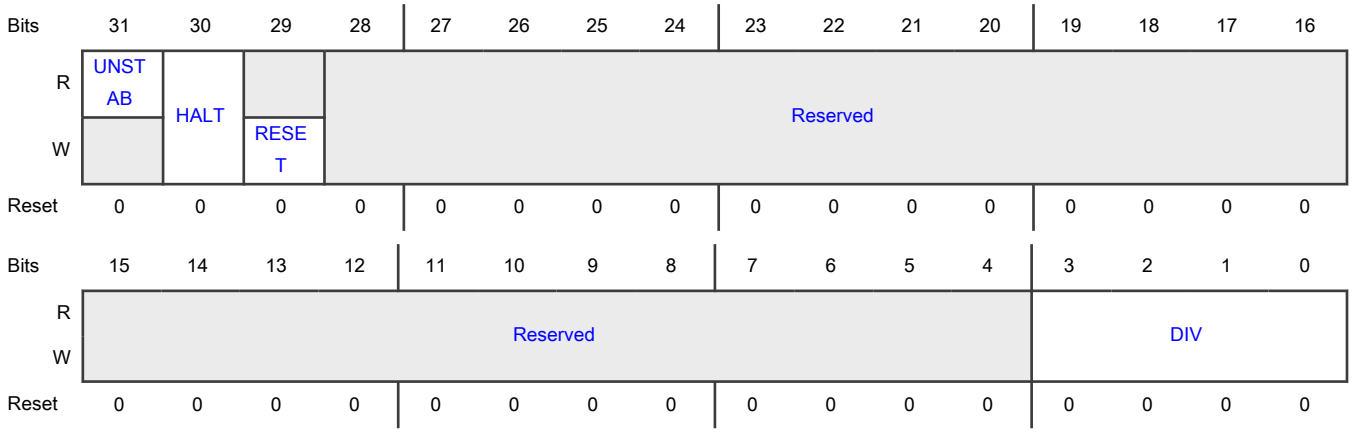
Offset

Register	Offset
MRCC_DBG_TRACE_CLKDIV	174h

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram



Fields

Field	Function
31 UNSTAB	Divider status flag Determines the stability of the divider clock. 0b - Divider clock is stable 1b - Clock frequency isn't stable
30 HALT	Halt divider counter Halt divider counter 0b - Divider clock is running 1b - Divider clock is stopped
29 RESET	Reset divider counter Reset divider counter 0b - Divider isn't reset 1b - Divider is reset
28-4 —	reserved reserved
3-0 DIV	Functional Clock Divider Before change peripheral's funcitional clock source, should configure the module's MRCC_GLB_CC and MRCC_GLB_ACC bits to 0. divider value = (DIV+1)

All information provided in this document is subject to legal disclaimers.

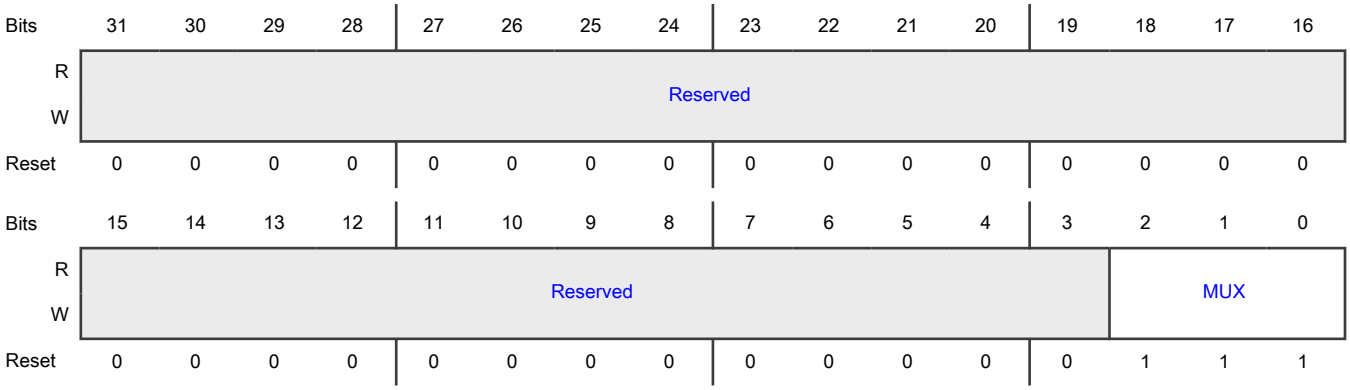
© 2026 NXP B.V. All rights reserved.

13.4.1.43 CLKOUT clock selection control (MRCC_CLKOUT_CLKSEL)

Offset

Register	Offset
MRCC_CLKOUT_CLKSEL	178h

Diagram



Fields

Field	Function
31-3 —	reserved reserved
2-0 MUX	Functional Clock Mux Select Before change peripheral's funcitional clock source, should configure the module's MRCC_GLB_CC and MRCC_GLB_ACC bits to 0. 000b - FRO_12M 001b - FRO_HF_DIV 010b - CLK_IN 011b - CLK_16K 110b - SLOW_CLK 111b - Reserved(NO Clock)

All information provided in this document is subject to legal disclaimers.

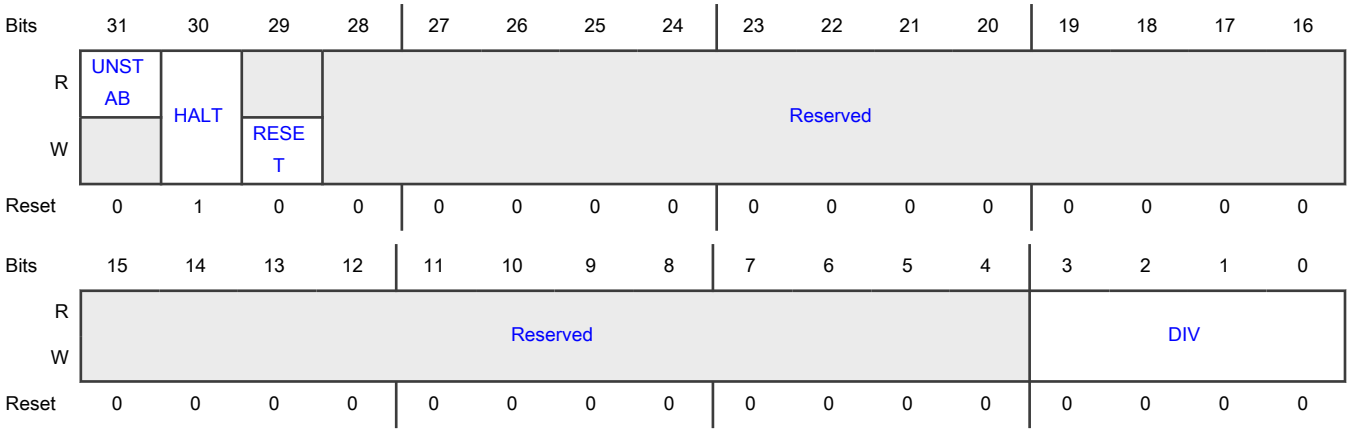
© 2026 NXP B.V. All rights reserved.

13.4.1.44 CLKOUT clock divider control (MRCC_CLKOUT_CLKDIV)

Offset

Register	Offset
MRCC_CLKOUT_CLKDIV	17Ch

Diagram



Fields

Field	Function
31 UNSTAB	Divider status flag Determines the stability of the divider clock. 0b - Divider clock is stable 1b - Clock frequency isn't stable
30 HALT	Halt divider counter Halt divider counter 0b - Divider clock is running 1b - Divider clock is stopped
29 RESET	Reset divider counter Reset divider counter 0b - Divider isn't reset 1b - Divider is reset
28-4 —	reserved reserved

Table continues on the next page...

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
3-0 DIV	Functional Clock Divider Before change peripheral's funcitional clock source, should configure the module's MRCC_GLB_CC and MRCC_GLB_ACC bits to 0. divider value = (DIV+1)

13.4.1.45 SYSTICK clock selection control (MRCC_SYSTICK_CLKSEL)

Offset

Register	Offset
MRCC_SYSTICK_CLKSEL	180h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved														MUX	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

Fields

Field	Function
31-2 —	reserved reserved
1-0 MUX	Functional Clock Mux Select Before change peripheral's funcitional clock source, should configure the module's MRCC_GLB_CC and MRCC_GLB_ACC bits to 0. 00b - CPU_CLK 01b - CLK_1M 10b - CLK_16K 11b - Reserved(NO Clock)

All information provided in this document is subject to legal disclaimers.

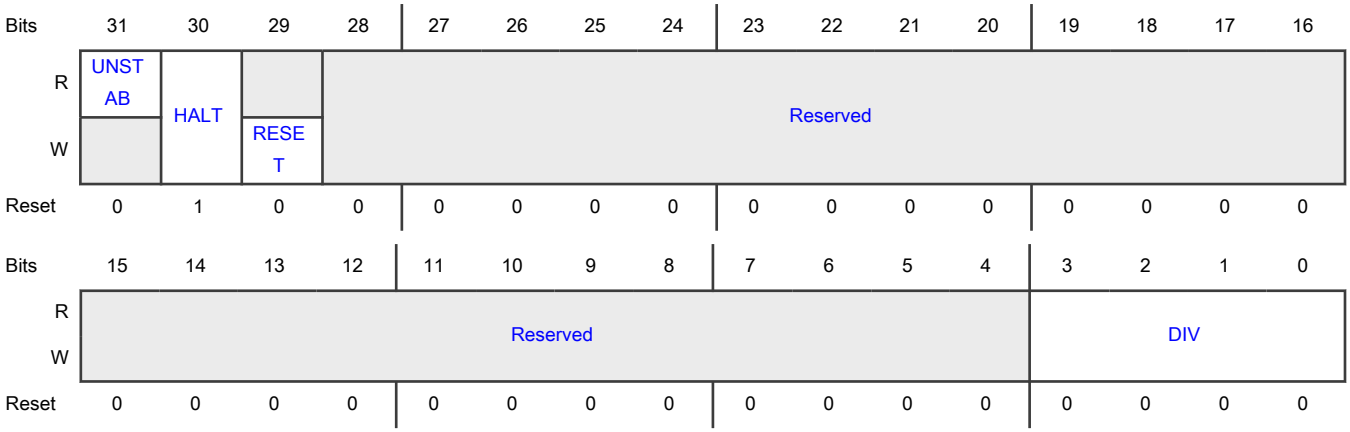
© 2026 NXP B.V. All rights reserved.

13.4.1.46 SYSTICK clock divider control (MRCC_SYSTICK_CLKDIV)

Offset

Register	Offset
MRCC_SYSTICK_CLKDIV	184h

Diagram



Fields

Field	Function
31 UNSTAB	Divider status flag Determines the stability of the divider clock. 0b - Divider clock is stable 1b - Clock frequency isn't stable
30 HALT	Halt divider counter Halt divider counter 0b - Divider clock is running 1b - Divider clock is stopped
29 RESET	Reset divider counter Reset divider counter 0b - Divider isn't reset 1b - Divider is reset
28-4 —	reserved reserved

Table continues on the next page...

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
3-0 DIV	Functional Clock Divider Before change peripheral's funcitional clock source, should configure the module's MRCC_GLB_CC and MRCC_GLB_ACC bits to 0. divider value = (DIV+1)

13.4.2 SECCON register descriptions

13.4.2.1 SECCON memory map

SYSCON__SECCON base address: 400E_8000h

Offset	Register	Width (In bits)	Access	Reset value
50h	CPU0 System Tick Calibration (CPU0STCKCAL)	32	RW	See section
61Ch	MSF Configuration (MSFCFG)	32	RW	0000_0000h
63Ch	CFPA State Register (CFPA_LC_STATE)	32	R	See section
658h	RAM XEN Control (RAM_XEN)	32	RW	0000_0000h
65Ch	RAM XEN Control (Duplicate) (RAM_XEN_DP)	32	RW	0000_0000h
680h	Life Cycle State Register (ELS_OTP_LC_STATE)	32	R	See section
684h	Life Cycle State Register (Duplicate) (ELS_OTP_LC_STATE_DP)	32	R	See section
7A0h	Control Write Access to Security (DEBUG_LOCK_EN)	32	RW	0000_000Ah
7A4h	Cortex Debug Features Control (DEBUG_FEATURES)	32	RW	See section
7A8h	Cortex Debug Features Control (Duplicate) (DEBUG_FEATURES_DP)	32	RW	See section
7B4h	CPU0 Software Debug Access (SWD_ACCESS_CPU0)	32	RW	0000_0000h

13.4.2.2 CPU0 System Tick Calibration (CPU0STCKCAL)

Offset

Register	Offset
CPU0STCKCAL	50h

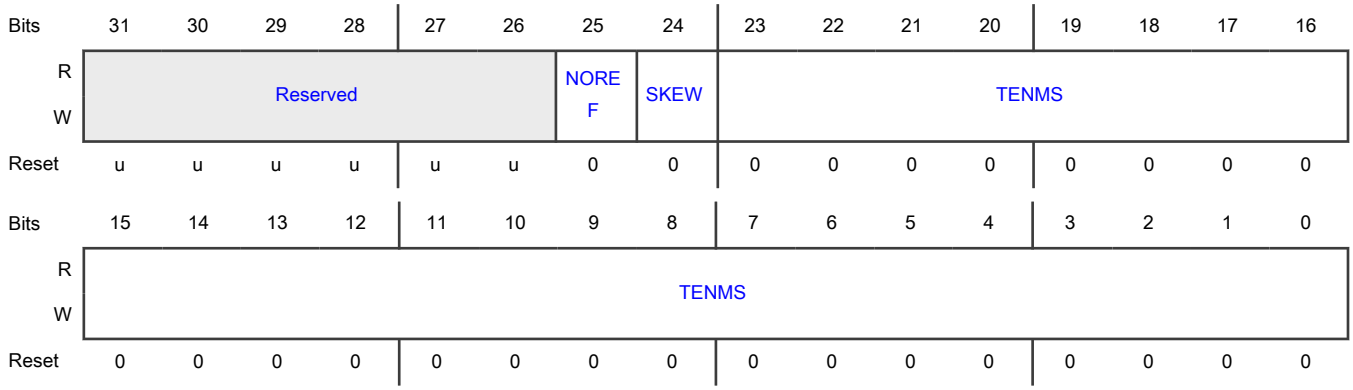
Function

The CPU0STCKCAL register allows software to set up a default value for the SYST_CALIB register (refer to Arm documentation) in the System Tick Timer of secure part of the CPU0.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram



Fields

Field	Function
31-26 —	Reserved
25 NOREF	Whether the device provides a reference clock to the processor. 0b - Reference clock is provided 1b - No reference clock is provided
24 SKEW	Whether the TENMS value is exact. 0b - TENMS value is exact 1b - TENMS value is not exact or not given
23-0 TENMS	Reload value for 10 ms (100 Hz) timing, subject to system clock skew errors. If the value reads as zero, the calibration value is not known.

13.4.2.3 MSF Configuration (MSFCFG)

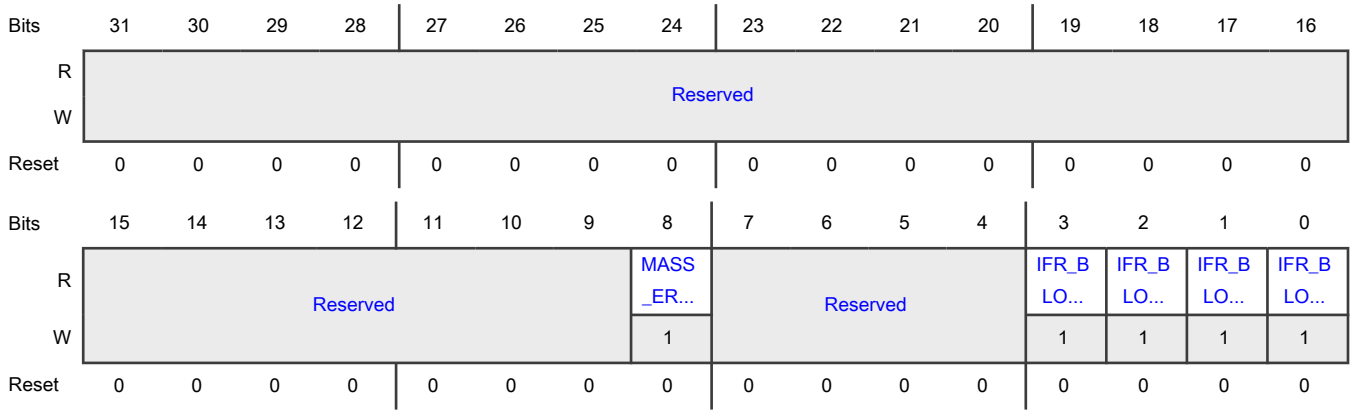
Offset

Register	Offset
MSFCFG	61Ch

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram



Fields

Field	Function
31-16 —	Reserved
15-9 —	Reserved
8 MASS_ERASE_DIS	Mass erase control 0b - Enables mass erase 1b - Disables mass erase, write one lock until a system reset.
7-4 —	Reserved
3-0 IFR_BLOCK0_ERASE_DISn	user IFR block0 sector n erase control 0b - Enable IFR sector erase. 1b - Disable IFR sector erase, write one lock until a system reset.

13.4.2.4 CFPA State Register (CFPA_LC_STATE)

Offset

Register	Offset
CFPA_LC_STATE	63Ch

Function

This register is written by Boot-ROM code and is only one-time write and locked down until a system reset.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	CFPA_LC_STATE															
W																
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CFPA_LC_STATE															
W																
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

Fields

Field	Function
31-0 CFPA_LC_STATE	CFPA state

13.4.2.5 RAM XEN Control (RAM_XEN)

Offset

Register	Offset
RAM_XEN	658h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	LOCK	Reserved														
W	1															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved								Reserved	Reserved	Reserved	Reserved	RAMA1_X...	RAMA0_X...	Reserved	Reserved
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Fields

Field	Function
31 LOCK	This 1-bit field provides a mechanism to limit writes to this register (and RAM_XEN_DP) to protect its contents. Once set, this bit remains asserted until a system reset. 0b - This register is not locked and can be altered. 1b - This register is locked and cannot be altered.
30-8 —	Reserved
7 —	Reserved
6 —	Reserved
5 —	Reserved
4 —	Reserved
3 RAMA1_XEN	RAMA1 Execute permission control. 0b - Execute permission is disabled, R/W are enabled. 1b - Execute permission is enabled, R/W/X are enabled.
2 RAMA0_XEN	RAMA0 Execute permission control. 0b - Execute permission is disabled, R/W are enabled. 1b - Execute permission is enabled, R/W/X are enabled.
1 —	Reserved
0 —	Reserved

13.4.2.6 RAM XEN Control (Duplicate) (RAM_XEN_DP)

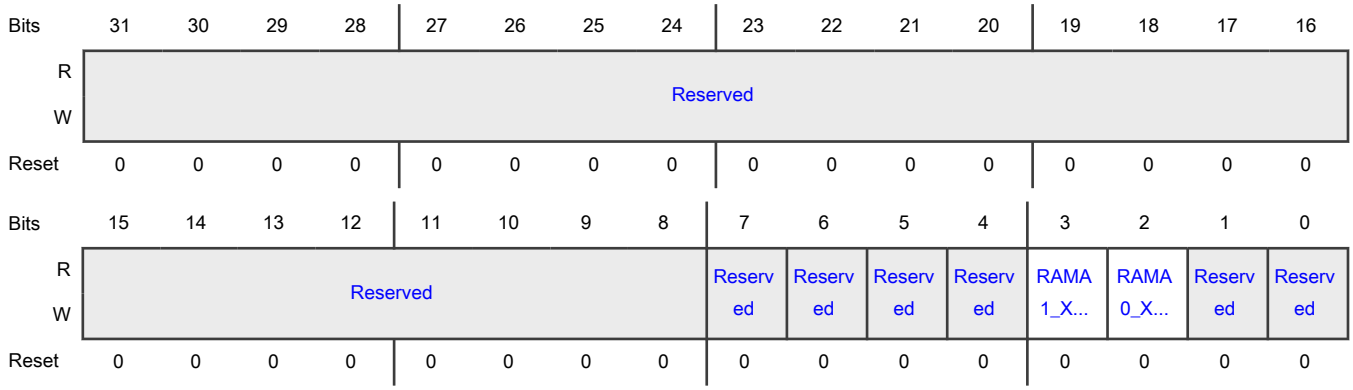
Offset

Register	Offset
RAM_XEN_DP	65Ch

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram



Fields

Field	Function
31-8 —	Reserved
7 —	Reserved
6 —	Reserved
5 —	Reserved
4 —	Reserved
3 RAMA1_XEN	Refer to RAM_XEN for more details.
2 RAMA0_XEN	Refer to RAM_XEN for more details.
1 —	Reserved
0 —	Reserved

All information provided in this document is subject to legal disclaimers.

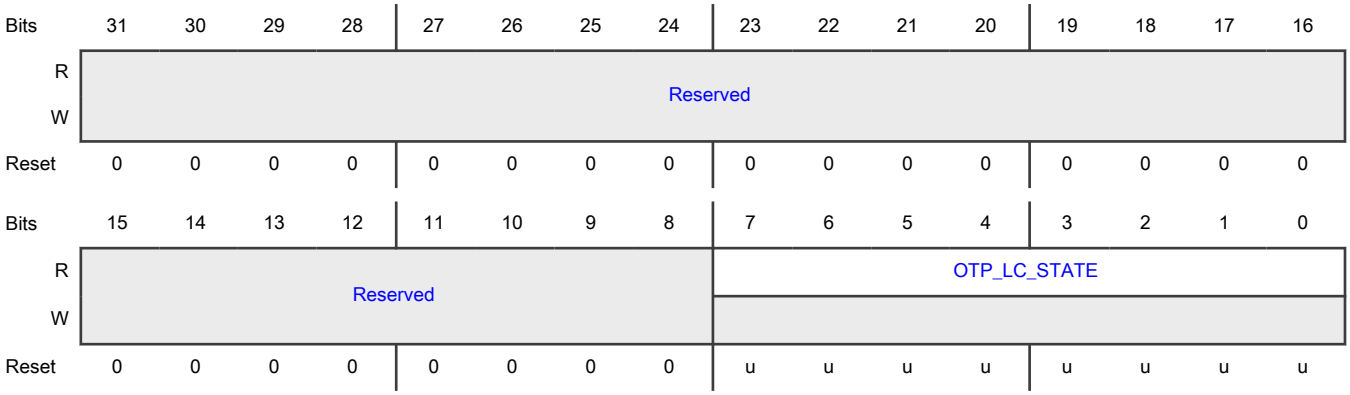
© 2026 NXP B.V. All rights reserved.

13.4.2.7 Life Cycle State Register (ELS_OTP_LC_STATE)

Offset

Register	Offset
ELS_OTP_LC_STATE	680h

Diagram



Fields

Field	Function
31-8 —	Reserved
7-0 OTP_LC_STATE	OTP life cycle state

13.4.2.8 Life Cycle State Register (Duplicate) (ELS_OTP_LC_STATE_DP)

Offset

Register	Offset
ELS_OTP_LC_STATE_DP	684h

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved								OTP_LC_STATE_DP							
W																
Reset	0	0	0	0	0	0	0	0	u	u	u	u	u	u	u	u

Fields

Field	Function
31-8 —	Reserved
7-0 OTP_LC_STATE_DP	OTP life cycle state

13.4.2.9 Control Write Access to Security (DEBUG_LOCK_EN)

Offset

Register	Offset
DEBUG_LOCK_EN	7A0h

Function

Controls write access to the SWD_ACCESS_CPU0, DEBUG_FEATURES, DEBUG_FEATURES_DP and DEBUG_AUTH0/1/2/3 registers

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved								Reserved				LOCK_ALL			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Fields

Field	Function
31-8 —	Reserved
7-4 —	Reserved
3-0 LOCK_ALL	Controls write access to the security registers Controls write access to the SWD_ACCESS_CPU0, DEBUG_FEATURES, DEBUG_FEATURES_DP registers 0000b - Any other value than b1010: disables write access to all registers 1010b - Enables write access to all registers

13.4.2.10 Cortex Debug Features Control (DEBUG_FEATURES)

Offset

Register	Offset
DEBUG_FEATURES	7A4h

Function

Cortex M33 (CPU0) debug features control.

NOTE

Reset on PoR only

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved		Reserved													
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		Reserved		Reserved				Reserved		Reserved		Reserved		CPU0_DBGEN	
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Fields

Field	Function
31-30 —	Reserved
29-14 —	Reserved
13-12 —	Reserved
11-8 —	Reserved
7-6 —	Reserved
5-4 —	Reserved
3-2 —	Reserved
1-0 CPU0_DBGEN	CPU0 invasive debug control Enables invasive debug for CPU0. Values not listed are reserved. 01b - Disables debug 10b - Enables debug

13.4.2.11 Cortex Debug Features Control (Duplicate) (DEBUG_FEATURES_DP)

Offset

Register	Offset
DEBUG_FEATURES_DP	7A8h

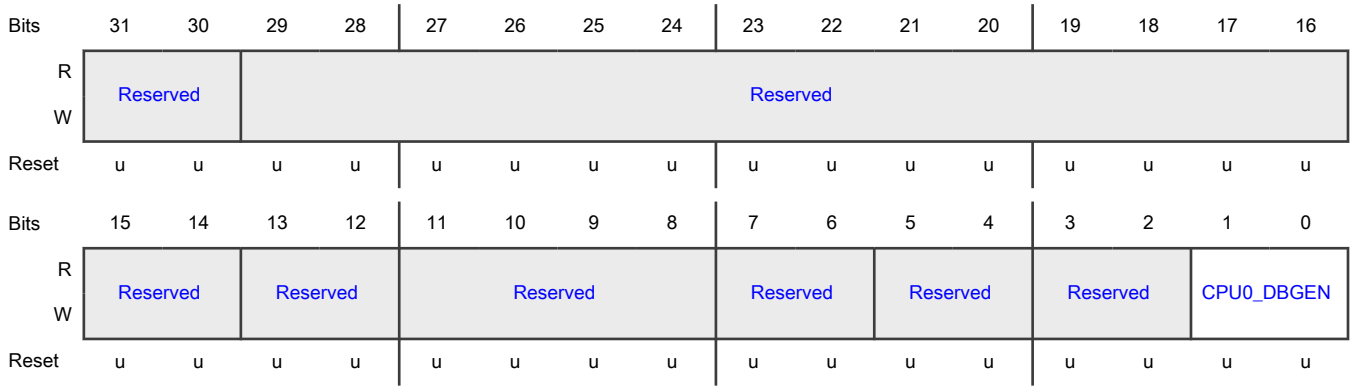
Function

Cortex M33 (CPU0) debug features control.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram



Fields

Field	Function
31-30 —	Reserved
29-14 —	Reserved
13-12 —	Reserved
11-8 —	Reserved
7-6 —	Reserved
5-4 —	Reserved
3-2 —	Reserved
1-0 CPU0_DBGEN	CPU0 invasive debug control Enables invasive debug for CPU0. Values not listed are reserved. 01b - Disables debug 10b - Enables debug

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

13.4.2.12 CPU0 Software Debug Access (SWD_ACCESS_CPU0)

Offset

Register	Offset
SWD_ACCESS_CPU0	7B4h

Function

This register is used by ROM during DEBUG authentication mechanism to enable debug access port for CPU0. write once only.

NOTE

Reset on PoR only

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SEC_CODE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SEC_CODE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-0 SEC_CODE	CPU0 SWD-AP: 0x12345678 0000_0000_0000_0000_0000_0000_0000_0000b - CPU0 DAP is not allowed. Reading back register is read as 0x5. 0001_0010_0011_0100_0101_0110_0111_1000b - Value to write to enable CPU0 SWD access. Reading back register is read as 0xA.

13.4.3 SYSCON register descriptions

13.4.3.1 SYSCON memory map

SYSCON__SYSCON base address: 4009_1000h

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Offset	Register	Width (In bits)	Access	Reset value
10h	AHB Matrix Priority Control (AHBMATPRIO)	32	RW	0000_0000h
48h	NMI Source Select (NMISRC)	32	RW	0000_0000h
4Ch	Protect Level Control (PROTLVL)	32	RW	0000_0000h
80h	System Clock Divider (AHBCLKDIV)	32	RW	0000_0000h
88h	SLOW_CLK Clock Divider (SLOWCLKDIV)	32	RW	0000_0003h
90h	FRO_HF_DIV Clock Divider (FROHFDIV)	32	RW	4000_0000h
94h	FRO_LF_DIV Clock Divider (FROLFDIV)	32	RW	4000_0000h
FCh	Clock Configuration Unlock (CLKUNLOCK)	32	RW	0000_0000h
200h	NVM Control (NVM_CTRL)	32	RW	0002_0400h
30Ch	CPU Status (CPUSTAT)	32	R	0000_0000h
438h	PWM0 Submodule Control (PWM0SUBCTL)	32	RW	0000_0000h
440h	CTIMER Global Start Enable (CTIMERGLOBALSTARTEN)	32	RW	0000_0000h
444h	RAM Control (RAM_CTRL)	32	RW	0000_0001h
7F0h	JTAG Chip ID (JTAG_ID)	32	R	0727_002Bh
7F4h	Device Type (DEVICE_TYPE)	32	R	See section
7F8h	Device ID (DEVICE_ID0)	32	R	See section

13.4.3.2 AHB Matrix Priority Control (AHBMATPRIO)

Offset

Register	Offset
AHBMATPRIO	10h

Function

The Multilayer AHB Matrix arbitrates between initiators, when they attempt to access the same matrix target port at the same time. The priority values are 3 = highest, 0 = lowest. When the priority is the same, the initiator with the lower initiator number is given priority.

NOTE

Be careful when modifying this register as improper settings can seriously degrade the performance.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved		Reserved		Reserved		Reserved		Reserved		Reserved		Reserved		Reserved	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		Reserved		Reserved		Reserved		Reserved		Reserved		DMA0		CPU0_BUS	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-30 —	Reserved
29-28 —	Reserved
27-26 —	Reserved
25-24 —	Reserved
23-22 —	Reserved
21-20 —	Reserved
19-18 —	Reserved
17-16 —	Reserved
15-14 —	Reserved
13-12 —	Reserved

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
11-10 —	Reserved
9-8 —	Reserved Read value is undefined, only zero should be written.
7-6 —	Reserved Read value is undefined, only zero should be written.
5-4 —	Reserved Read value is undefined, only zero should be written.
3-2 DMA0	DMA0 controller bus initiator priority level 00b - level 0 01b - level 1 10b - level 2 11b - level 3
1-0 CPU0_BUS	CPU0 AHB bus initiator priority level 00b - level 0 01b - level 1 10b - level 2 11b - level 3

13.4.3.3 NMI Source Select (NMISRC)

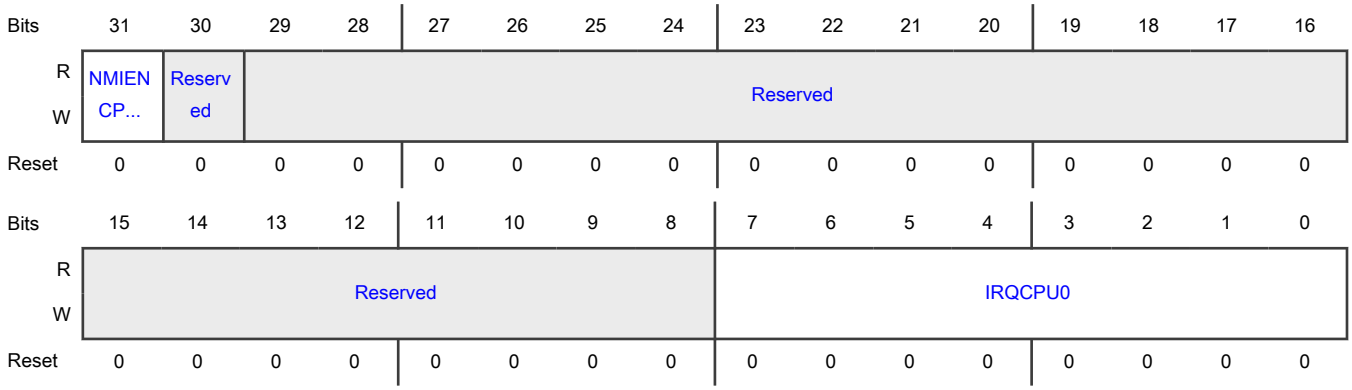
Offset

Register	Offset
NMISRC	48h

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram



Fields

Field	Function
31 NMIENCPU0	Enables the Non-Maskable Interrupt (NMI) source selected by IRQCPU0. 0b - Disable. 1b - Enable.
30 —	Reserved
29-16 —	Reserved
15-8 —	Reserved
7-0 IRQCPU0	The IRQ number of the interrupt that acts as the Non-Maskable Interrupt (NMI) for CPU0, if enabled by NMIENCPU0.

13.4.3.4 Protect Level Control (PROTLVL)

Offset

Register	Offset
PROTLVL	4Ch

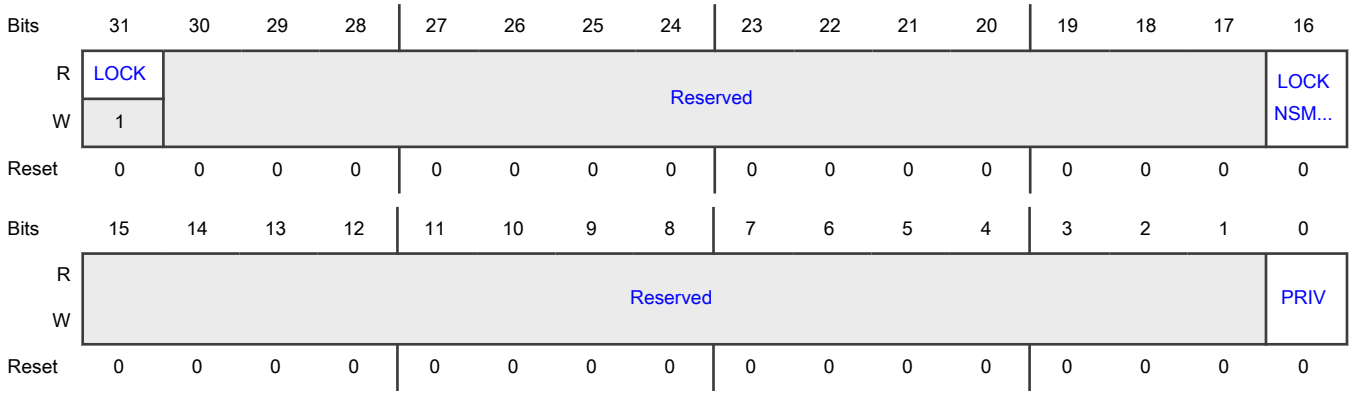
Function

This register is privileged access only.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram



Fields

Field	Function
31 LOCK	This 1-bit field provides a mechanism to limit writes to this register to protect its contents. Once set, this bit remains asserted until a system reset. 0b - This register is not locked and can be altered. 1b - This register is locked and cannot be altered until a system reset.
30-17 —	Reserved
16 LOCKNSMPU	Control write access to Nonsecure MPU memory regions. 0b - Unlock these registers. privileged access to Nonsecure MPU memory regions is allowed. 1b - Disable writes to the MPU_CTRL_NS, MPU_RNR_NS, MPU_RBAR_NS, MPU_RLAR_NS, MPU_RBAR_A_NSn and MPU_RLAR_A_NSn. All writes to the registers are ignored.
15-1 —	Reserved
0 PRIV	Control privileged access of EIM, ERM, Flexcan, MBC, SCG, DMA, ROMCP and Flexspi. 0b - privileged access is disabled. the peripherals could be access in user mode. 1b - privileged access is enabled. the peripherals could be access in privilege mode.

13.4.3.5 System Clock Divider (AHBCLKDIV)

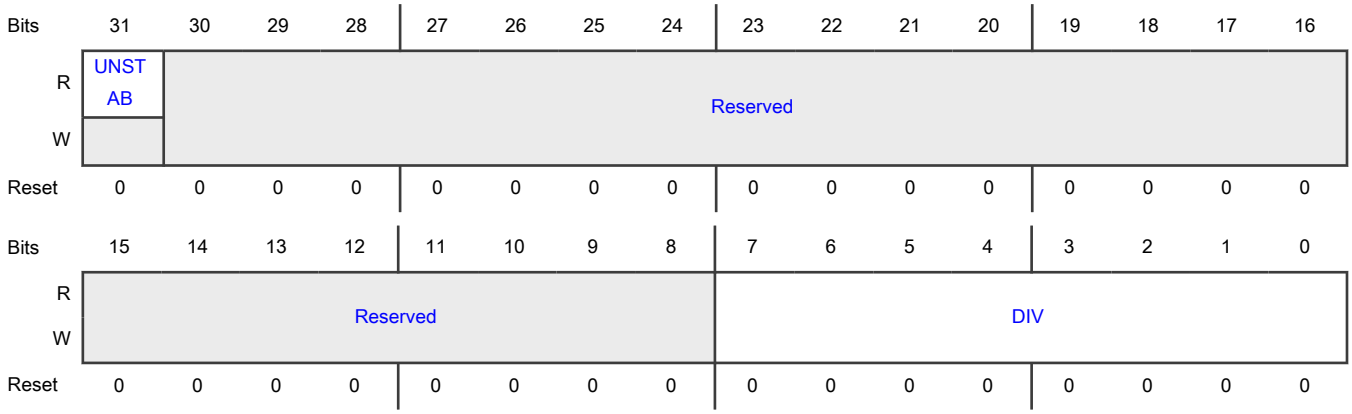
Offset

Register	Offset
AHBCLKDIV	80h

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram



Fields

Field	Function
31 UNSTAB	Divider status flag 0b - Divider clock is stable 1b - Clock frequency is not stable
30-8 —	Reserved
7-0 DIV	Clock divider value The divider value = (DIV + 1)

13.4.3.6 SLOW_CLK Clock Divider (SLOWCLKDIV)

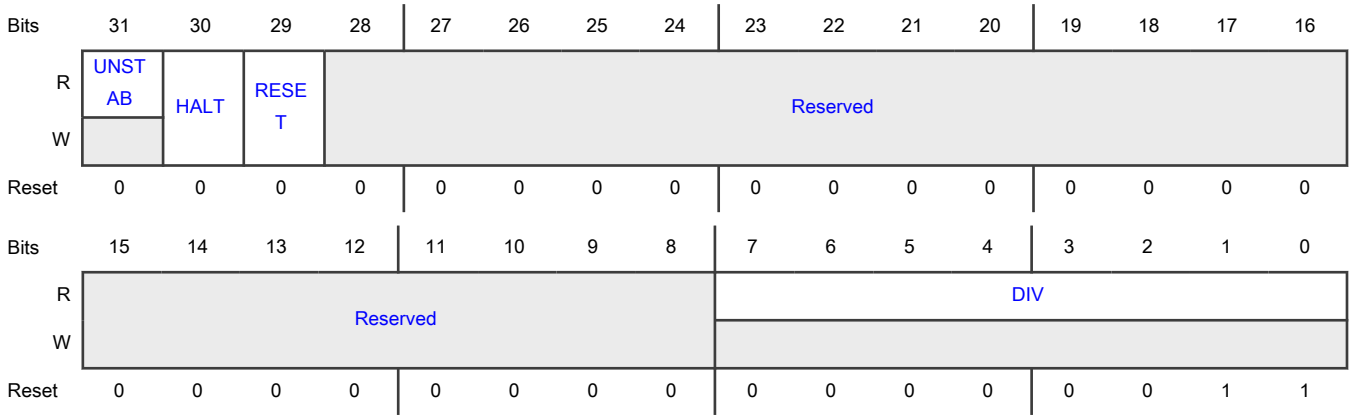
Offset

Register	Offset
SLOWCLKDIV	88h

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram



Fields

Field	Function
31 UNSTAB	Divider status flag 0b - Divider clock is stable 1b - Clock frequency is not stable
30 HALT	Halts the divider counter 0b - Divider clock is running 1b - Divider clock is stopped
29 RESET	Resets the divider counter **NOTE** Limited to configure RESET bit when HALT bit is set to 1 0b - Divider is not reset 1b - Divider is reset
28-8 —	Reserved
7-0 DIV	Clock divider value The divider value = (DIV + 1)

13.4.3.7 FRO_HF_DIV Clock Divider (FROHFDIV)

Offset

Register	Offset
FROHFDIV	90h

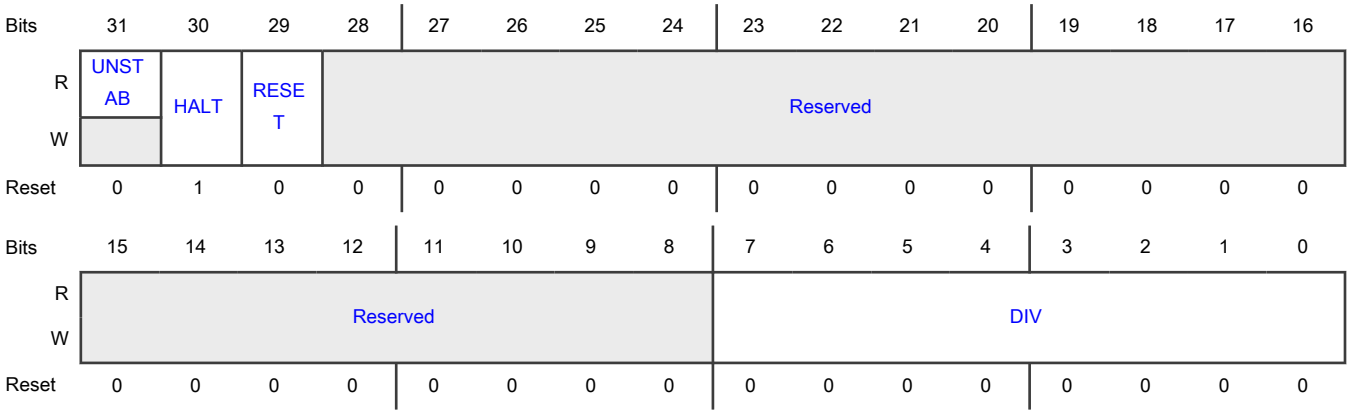
All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Function

This register is used to generate fro_hf_div clock from fro_hf clock.

Diagram



Fields

Field	Function
31 UNSTAB	Divider status flag 0b - Divider clock is stable 1b - Clock frequency is not stable
30 HALT	Halts the divider counter 0b - Divider clock is running 1b - Divider clock is stopped
29 RESET	Resets the divider counter 0b - Divider is not reset 1b - Divider is reset
28-8 —	Reserved
7-0 DIV	Clock divider value The divider value = (DIV + 1)

13.4.3.8 FRO_LF_DIV Clock Divider (FROLFDIV)

Offset

Register	Offset
FROLFDIV	94h

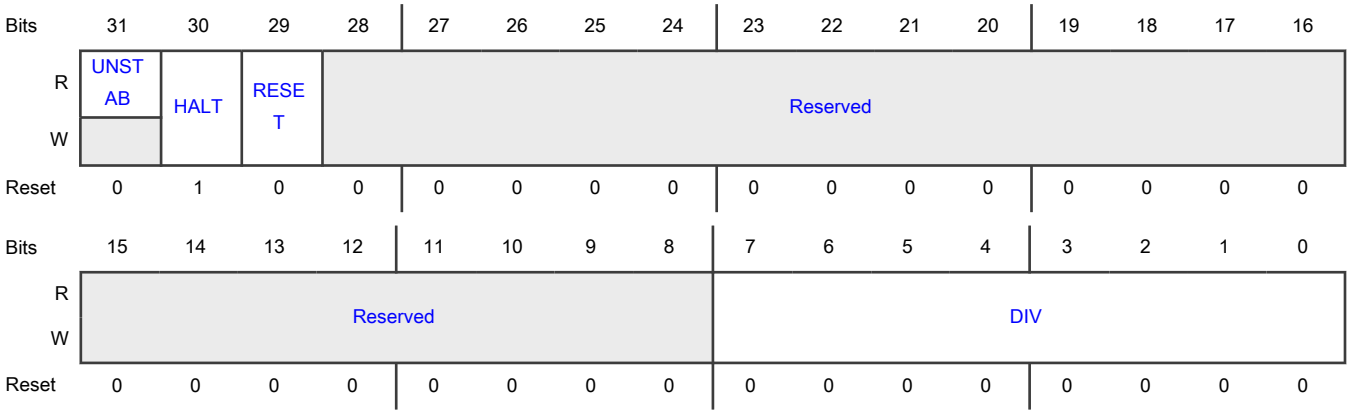
All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Function

This register is used to generate fro_lf_div clock from fro_lf clock.

Diagram



Fields

Field	Function
31 UNSTAB	Divider status flag 0b - Divider clock is stable 1b - Clock frequency is not stable
30 HALT	Halts the divider counter 0b - Divider clock is running 1b - Divider clock is stopped
29 RESET	Resets the divider counter 0b - Divider is not reset 1b - Divider is reset
28-8 —	Reserved
7-0 DIV	Clock divider value The divider value = (DIV + 1)

13.4.3.9 Clock Configuration Unlock (CLKUNLOCK)

Offset

Register	Offset
CLKUNLOCK	FCh

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Function

This register controls access to the clock select and divider configuration registers.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															LOCK
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-1 —	Reserved
0 LOCK	Controls clock configuration registers access (for example, SLOWCLKDIV, BUSCLKDIV, AHBCLKDIV, FROHFDIV, FROLFDIV, PLLxCLKDIV, MRCC_xxx_CLKDIV, MRCC_xxx_CLKSEL, MRCC_GLB_xxx) 0b - Updates are allowed to all clock configuration registers. 1b - Freezes all clock configuration registers update.

13.4.3.10 NVM Control (NVM_CTRL)

Offset

Register	Offset
NVM_CTRL	200h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved														DIS_MBE...	DIS_MBE...
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved					FLASH_S...	Reserved		Reserved				Reserved		DIS_DAT...	DIS_FLA...
Reset	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Fields

Field	Function
31-18 —	Reserved
17 DIS_MBECC_ERR_DATA	Bus error on data multi-bit ECC error control **NOTE** Set this field to 0 if you want to enable flash speculative 0b - Enables bus error on multi-bit ECC error for data 1b - Disables bus error on multi-bit ECC error for data
16 DIS_MBECC_ERR_INST	Bus error on data multi-bit ECC error control **NOTE** Set this field to 0 if you want to enable flash speculative 0b - Enables bus error on multi-bit ECC error for instruction 1b - Disables bus error on multi-bit ECC error for instruction
15-11 —	Reserved
10 FLASH_STALL_EN	FLASH stall on busy control 0b - No stall on FLASH busy 1b - Stall on FLASH busy
9 —	Reserved Keep the default value.
8-6 —	Reserved
5-2 —	Reserved
1 DIS_DATA_SPEC	Flash data speculation control **NOTE** If DIS_MBECC_ERR_DATA and/or DIS_MBECC_ERR_INST are set, then speculation will not be enabled, even if this bit is cleared. 0b - Enables data speculation 1b - Disables data speculation
0	Flash speculation control

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

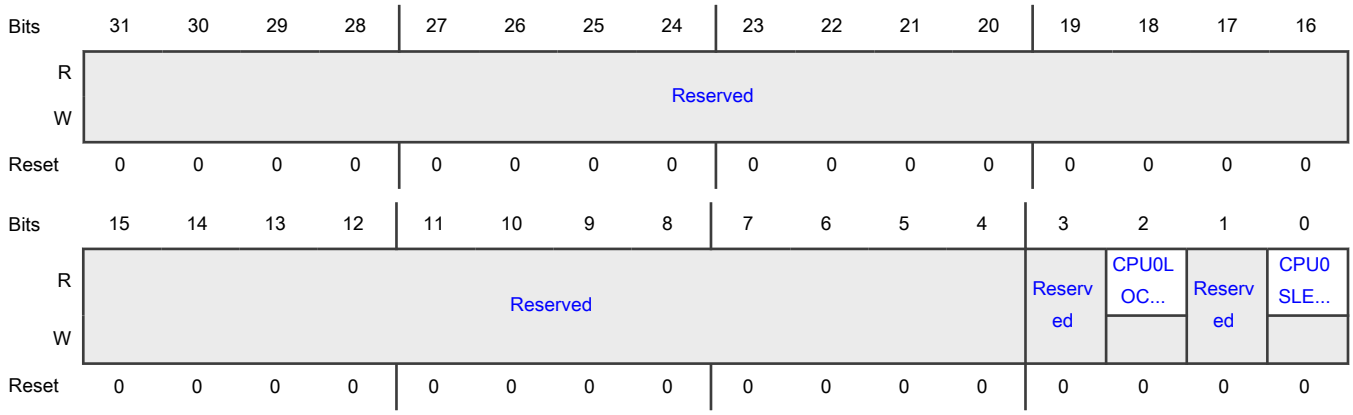
Field	Function
DIS_FLASH_SPEC	**NOTE** If DIS_MBECC_ERR_DATA and/or DIS_MBECC_ERR_INST are set, then speculation will not be enabled, even if this bit is cleared. 0b - Enables flash speculation 1b - Disables flash speculation

13.4.3.11 CPU Status (CPUSTAT)

Offset

Register	Offset
CPUSTAT	30Ch

Diagram



Fields

Field	Function
31-4 —	Reserved
3 —	Reserved
2 CPU0LOCKUP	CPU0 lockup state 0b - CPU is not in lockup 1b - CPU is in lockup

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
1 —	Reserved
0 CPU0SLEEPING	CPU0 sleeping state 0b - CPU is not sleeping 1b - CPU is sleeping

13.4.3.12 PWM0 Submodule Control (PWM0SUBCTL)

Offset

Register	Offset
PWM0SUBCTL	438h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R																
W	0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R													CLK3_EN	CLK2_EN	CLK1_EN	CLK0_EN
W	0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-16 —	Reserved
15-4 —	Reserved
3-0 CLKn_EN	Enables PWM0 SUB Clockn 0b - Disable 1b - Enable

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

13.4.3.13 CTIMER Global Start Enable (CTIMERGLOBALSTARTEN)

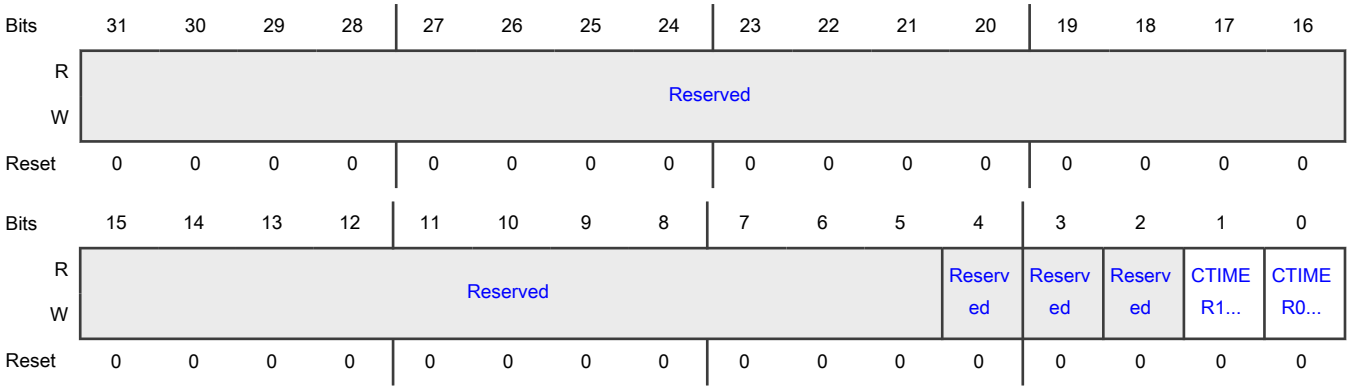
Offset

Register	Offset
CTIMERGLOBALSTARTEN	440h

Function

Refer to CTIMERn TCR.AGCEN

Diagram



Fields

Field	Function
31-5 —	Reserved
4 —	Reserved
3 —	Reserved
2 —	Reserved
1 CTIMER1_CLK_EN	Enables the CTIMER1 function clock 0b - Disable 1b - Enable
0 CTIMER0_CLK_EN	Enables the CTIMER0 function clock 0b - Disable 1b - Enable

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

13.4.3.14 RAM Control (RAM_CTRL)

Offset

Register	Offset
RAM_CTRL	444h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved								Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	RAMA_CG...
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved								Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	RAMA_EC...
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Fields

Field	Function
31-24 —	Reserved
23 —	Reserved
22 —	Reserved
21 —	Reserved
20 —	Reserved
19 —	Reserved
18 —	Reserved
17	Reserved

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
—	
16 RAMA_CG_OVERRIDE	RAMA bank clock gating control, only avaiable when RAMA_ECC_ENABLE = 0. 0b - Memory bank clock is gated automatically if no access more than 16 clock cycles. Auto-clkgating using bank_adapter gives a timeout counter for each bank. Once it expires upon no access per specified width (16 cycles), then clock to the bank is stopped high and one cycle penalty will be incurred if a read is the first access after the break. A write will incur no penalty. 1b - Auto clock gating feature is disabled
15-8 —	Reserved
7 —	Reserved
6 —	Reserved
5 —	Reserved
4 —	Reserved
3 —	Reserved
2 —	Reserved
1 —	Reserved
0 RAMA_ECC_ENABLE	RAMA ECC enable 0b - ECC is disabled 1b - ECC is enabled

13.4.3.15 JTAG Chip ID (JTAG_ID)

Offset

Register	Offset
JTAG_ID	7F0h

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	JTAG_ID															
W																
Reset	0	0	0	0	0	1	1	1	0	0	1	0	0	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	JTAG_ID															
W																
Reset	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	1

Fields

Field	Function
31-0 JTAG_ID	Indicates the device ID

13.4.3.16 Device Type (DEVICE_TYPE)

Offset

Register	Offset
DEVICE_TYPE	7F4h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	DEVICE_TYPE_PIN								DEVICE_TYPE_PKG				Reserved			DEVICE_...
W																
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DEVICE_TYPE_NUM															
W																
Reset	0	1	0	1	u	u	u	u	u	u	u	u	u	u	u	u

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Fields

Field	Function
31-24 DEVICE_TYPE_PIN	Indicates the device's pin number
23-20 DEVICE_TYPE_PKG	Indicates the device's package type 0000b - HLQFP 0001b - HTQFP 0010b - BGA 0011b - HDQFP 0100b - QFN 0101b - CSP 0110b - LQFP
19-17 —	Reserved
16 DEVICE_TYPE_SEC	Indicates the device type 0b - Non Secure 1b - Secure
15-0 DEVICE_TYPE_NUM	Indicates the device part number [15:12]: ... 0101b: Cxxx ... [11:0]: 3 digital of the part unmber. For example: C151: 0x5151 C162: 0x5162

13.4.3.17 Device ID (DEVICE_ID0)

Offset

Register	Offset
DEVICE_ID0	7F8h

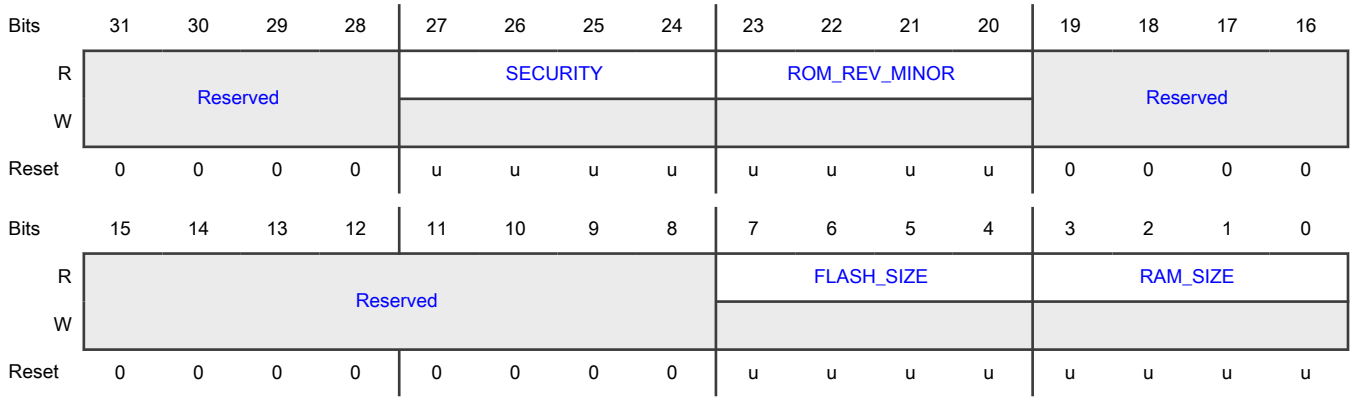
Function

This register contains the device ID.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram



Fields

Field	Function
31-28 —	Reserved
27-24 SECURITY	0101b - Secure version. (All values other than 1010b represent the secure version.) 1010b - Non secure version.
23-20 ROM_REV_MINOR	Indicates the device's ROM revision
19-8 —	Reserved
7-4 FLASH_SIZE	Indicates the device's flash size 0000b - 32KB. 0001b - 64KB. 0010b - 128KB. 0011b - 256KB. 0100b - 512KB. 0101b - 768KB. 0110b - 1MB. 0111b - 1.5MB. 1000b - 2MB.
3-0 RAM_SIZE	Indicates the device's ram size 0000b - 6KB. 0001b - 12KB.

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	0010b - 32KB. 0011b - 64KB. 0100b - 96KB. 0101b - 128KB. 0110b - 160KB. 0111b - 192KB. 1000b - 256KB. 1001b - 288KB. 1010b - 352KB. 1011b - 512KB. 1100b - 640KB.

13.5 Application information

Peripheral clock is gated by using CC (CC = {MRCC_GLB_ACC.*[peripherals name]*, MRCC_GLB_CC.*[peripherals name]* })

1. CC = 0b00 : Peripheral clocks are disabled; module does not stall low power mode entry. Clock is always gated, peripheral access causes hard fault.
2. CC = 0b01 : Peripheral clocks are enabled; module does not stall low power mode entry. Clock is never gated unless root clock source is gated in DEEPSLEEP mode.
3. CC = 0b10 : Peripheral clocks are enabled unless module is idle; low power mode entry stalls until module is idle. Only LPUART, LPSPI, LPI2C, FLEXIO, GPIO, FMU support CC=0b10. for other modules, it's same as CC=0b11.
4. CC = 0b11 : Peripheral clocks are enabled unless in DEEPSLEEP (or lower) mode; low power mode entry stalls until module is idle.

NOTE

FMC's CC bits are only available during low power mode, FMC is accessible and no hardfault occurs if CC=0b00 in RUN mode.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Chapter 14
Enhanced DMA Controller (eDMA3)

14.1 Chip-specific eDMA information

Table 75. Reference links to related information

Topic	Related module	Reference
Full description	eDMA	DMA3
System memory map		Memory map
Clocking		Clock distribution
Power Management		Power Management
Signal multiplexing	Port control	Signal multiplexing

14.1.1 Module instances

This chip has one instance of 4-channel eDMA module.

14.1.2 Direct Memory Access Multiplexer (DMAMUX)

The peripheral request line for each channel of the Direct Memory Access Controller (DMA) is driven from a Direct Memory Access Multiplexer (DMAMUX). This is a flexible configuration that allows the user to select the appropriate module to connect to each channel of the DMA Controller. The DMAMUX for this device allows DMA request signals to be mapped to each channel. Because of the mux, there is not a correlation between any of the DMA request sources and a specific DMA channel.

The DMAMUX is an integrated component of the DMA Controller.

14.1.2.1 DMAMUX0 request assignments

The following table provides DMA channel and source assignments for DMAMUX0.

Table 76. DMAMUX0 request assignments

Slot	Module	DMA request	Async DMA (Y/N)
0	—	Disabled	—
1	WUU0	Wake up event	Y
2	Reserved	Reserved	—
3	Reserved	Reserved	—
4	Reserved	Reserved	—
5	Reserved	Reserved	—
6	Reserved	Reserved	—
7	Reserved	Reserved	—
8	Reserved	Reserved	—
9	Reserved	Reserved	—
10	Reserved	Reserved	—

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.
© 2026 NXP B.V. All rights reserved.

Table 76. DMAMUX0 request assignments (continued)

Slot	Module	DMA request	Async DMA (Y/N)
11	LPI2C0	Receive request	Y
12	LPI2C0	Transmit request	Y
13	Reserved	Reserved	—
14	Reserved	Reserved	—
15	LPSPI0	Receive request	Y
16	LPSPI0	Transmit request	Y
17	Reserved	Reserved	—
18	Reserved	Reserved	—
19	Reserved	Reserved	—
20	Reserved	Reserved	—
21	LPUART0	Receive request	Y
22	LPUART0	Transmit request	Y
23	LPUART1	Receive request	Y
24	LPUART1	Transmit request	Y
25	LPUART2	Receive request	Y
26	LPUART2	Transmit request	Y
27	LPUART3	Receive request	Y
28	LPUART3	Transmit request	Y
29	Reserved	Reserved	—
30	Reserved	Reserved	—
31	CTIMER0	DMAREQ_M0	N
32	CTIMER0	DMAREQ_M1	N
33	CTIMER1	DMAREQ_M0	N
34	CTIMER1	DMAREQ_M1	N
35	Reserved	Reserved	—
36	Reserved	Reserved	—
37	Reserved	Reserved	—
38	Reserved	Reserved	—
39	Reserved	Reserved	—
40	Reserved	Reserved	—
41	FlexPWM0	Req_capt0	N
42	FlexPWM0	Req_capt1	N

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table 76. DMAMUX0 request assignments (continued)

Slot	Module	DMA request	Async DMA (Y/N)
43	FlexPWM0	Req_capt2	N
44	Reserved	Reserved	—
45	FlexPWM0	Req_val0	N
46	FlexPWM0	Req_val1	N
47	FlexPWM0	Req_val2	N
48	Reserved	Reserved	—
49	LPTMR0	Counter match event	Y
50	Reserved	Reserved	—
51	ADC0	FIFO Request	Y
52	Reserved	Reserved	—
53	LPCMP0	DMA_request	Y
54	Reserved	Reserved	—
55	Reserved	Reserved	—
56	Reserved	Reserved	—
57	Reserved	Reserved	—
58	Reserved	Reserved	—
59	Reserved	Reserved	—
60	GPIO0	Pin event request 0	Y
61	GPIO1	Pin event request 0	Y
62	GPIO2	Pin event request 0	Y
63	GPIO3	Pin event request 0	Y

14.2 Overview

The enhanced direct memory access (eDMA) controller is capable of performing complex data transfers with minimal intervention from a host processor. The hardware microarchitecture includes:

- A DMA engine that performs:
  - Source address and destination address calculations
  - Data-movement operations
- Local memory containing transfer control descriptors for each of the 4 channels

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

NOTE

The terminology in this chapter has been updated to align with NXP's inclusive language standards, as shown in the table below.

Table 77. Updated terms

Updated term	Deprecated term
Initiator	Master
Target	Slave

14.2.1 Block diagram

Figure 17 illustrates the components of the eDMA system, including the eDMA module (engine).

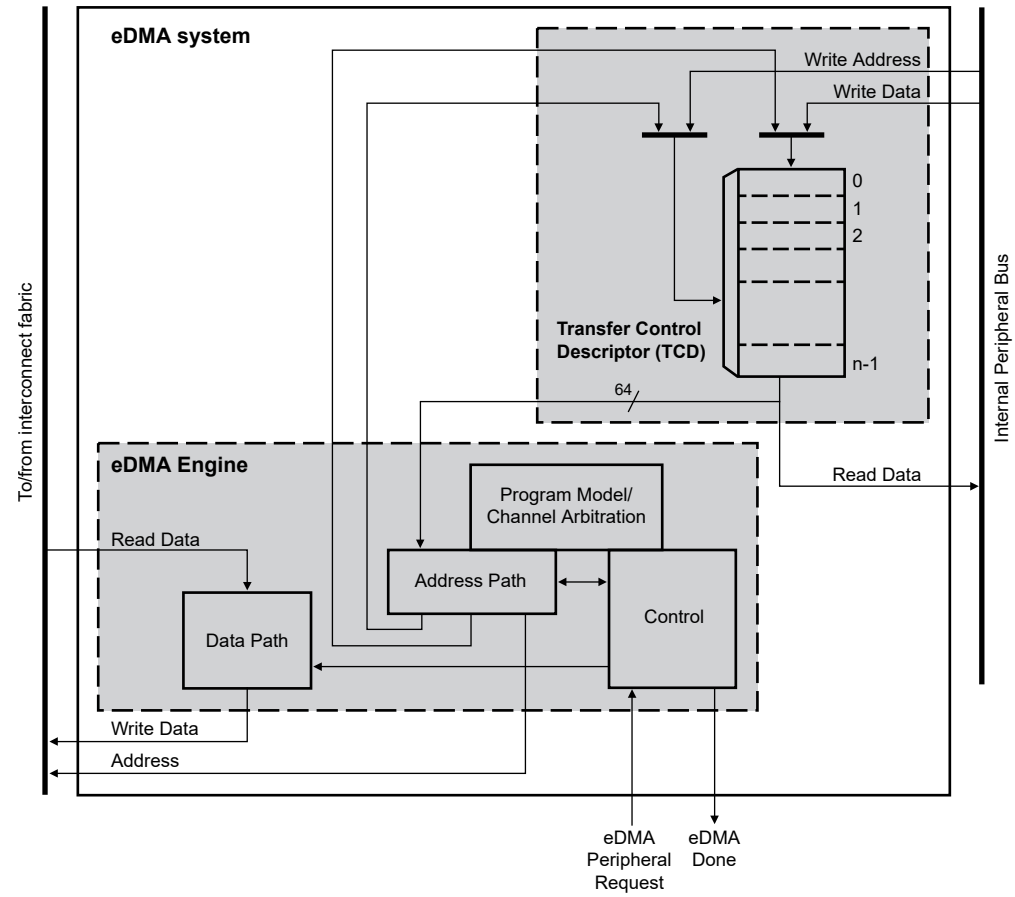


Figure 17. Block diagram

14.2.2 Block parts

The eDMA module is partitioned into two major modules: the eDMA engine and the transfer control descriptor local memory.

The eDMA engine is further partitioned into four submodules:

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table 78. eDMA engine submodules

Submodule	Function
Address path	This block: • Implements a primary channel and secondary (preempt) channel • Manages all initiator bus-address calculations All the channels provide the same functionality. This structure allows data transfers associated with one channel to be preempted after the completion of a read/write sequence if a higher priority channel activation is asserted while the primary channel is active. After a channel is activated, it runs until the minor loop is completed, unless preempted by a higher priority channel. This provides a mechanism (enabled by CH*n*_PRI[ECP]) where a large data transfer can be preempted to minimize the time another channel is blocked from execution. When any channel is selected to execute, the contents of its TCD are read from local memory and loaded into the address path channel x registers for a normal start and into channel y registers for a preemption start. After the minor loop completes execution, the address path hardware writes the new values for the TCD*n*_{SADDR, DADDR, CITER} back to local memory. If the major iteration count is exhausted, additional processing is performed, including the final address pointer updates, reloading the TCD*n*_CITER field, and a possible fetch of a new TCD*n* from memory as part of a scatter/gather operation. See Dynamic scatter/gather for more details.
Data path	This block implements the bus initiator read/write data path. It includes a data buffer and the necessary multiplex logic to support any required data alignment. The internal read data bus is the primary input, and the internal write data bus is the primary output. The address and data path modules directly support the 2-stage pipelined internal bus. The address path module represents the first stage of the bus pipeline (address phase), and the data path module implements the second stage of the pipeline (data phase).
Program model/channel arbitration	This block implements the first section of the eDMA programming model as well as the channel arbitration logic. The programming model registers are connected to the internal peripheral bus. The eDMA peripheral request inputs and interrupt request outputs are also connected to this block (via control logic).
Control	This block provides all the control functions for the eDMA engine. For data transfers where the source and destination sizes are equal, the eDMA engine performs a series of source read/destination write operations until the number of bytes specified in the minor loop byte count has been moved from the source to the destination. For descriptors where the sizes are not equal, multiple accesses of the smaller size data are required for each reference of the larger size. As an example, if the source size references 16-bit data and the destination is 32-bit data, the eDMA performs two reads, then one 32-bit write.

The transfer control descriptor local memory is further partitioned into:

Table 79. Transfer control descriptor memory

Submodule	Description
Memory controller	This logic implements the required dual-ported controller, and manages accesses from the eDMA engine as well as references from the internal peripheral bus. In simultaneous accesses, the eDMA engine is given priority and the peripheral transaction is stalled.
Memory array	TCD storage for each channel's transfer profile.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

14.2.3 Features

The eDMA is a highly programmable data-transfer engine optimized to minimize any required intervention from the host processor. It is intended for use in applications where the data size to be transferred is statically known and is not defined within the transferred data itself. The eDMA module features:

- All data movement via dual-address transfers: read from source, write to destination
  - Programmable source and destination addresses and transfer size
  - Support for complex address calculations
- 4-channel implementation that performs complex data transfers with minimal intervention from a host processor
  - Internal data buffer, used as temporary storage for all transfers
  - Connections to the crossbar switch for bus driving the data movement
- TCD organized to support two-deep, nested transfer operations
  - 32-byte TCD stored in local memory for each channel
  - An inner data transfer loop defined by a minor byte transfer count
  - An outer data transfer loop defined by a major iteration count
- Channel activation via one of three methods:
  - Explicit software initiation
  - Initiation via a channel-to-channel linking mechanism for continuous transfers
  - Peripheral-paced hardware requests, one per channel
- Fixed-priority and round-robin channel arbitration
- Channel completion reported via programmable interrupt requests
  - One interrupt per channel, which can be asserted at completion of major iteration count
  - Programmable error terminations per channel that are logically summed together to form one error interrupt to the interrupt controller
- Programmable support for scatter/gather DMA processing
- Support for complex data structures

In the discussion of this module, *n* is used to reference the channel number.

14.3 Functional description

The operation of eDMA is described in the following subsections.

14.3.1 Modes of operation

The eDMA operates in the following modes:

Table 80. Modes of operation

Mode	Description
Normal	In Normal mode, eDMA transfers data between a source and a destination. The source and destination can be a memory block or an I/O block capable of operation with eDMA. A service request initiates a transfer of a specific number of bytes (NBYTES) as specified in the TCD. The minor loop is the sequence of read-write operations that transfers these NBYTES per service

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table 80. Modes of operation (continued)

Mode	Description
	request. Each service request executes one iteration of the major loop, which transfers NBYTES of data.
Debug	eDMA operation is configurable in Debug mode via the control register: • If CSR[EDBG] is cleared to 0, eDMA continues to operate. • If CSR[EDBG] is set to 1, eDMA stops transferring data. If Debug mode is entered when a channel is active, eDMA continues operation until the channel retires.

14.3.2 eDMA basic data flow

The basic flow of a data transfer can be partitioned into three segments.

As shown in the following diagram, the first segment involves the channel activation:

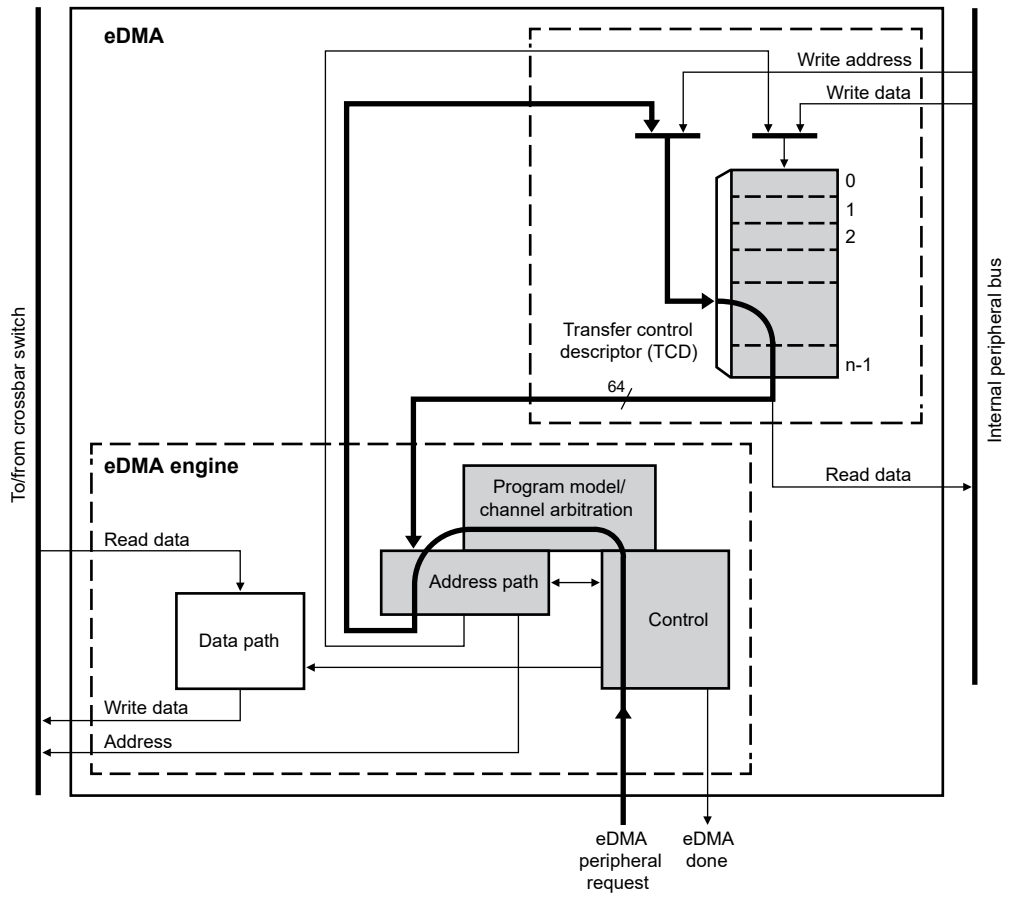


Figure 18. eDMA operation, part 1

This example uses the assertion of the eDMA peripheral request signal to request service for channel *n*. Channel activation via software and the TCD*n*_CSR[START] field follows the same basic flow as peripheral requests. The eDMA request input signal is registered internally and then routed through the eDMA engine: first through the control module, then into the program model and channel arbitration.

In the next cycle, the channel arbitration begins using fixed-priority plus the optional round-robin algorithm. After arbitration is complete, the activated channel number is sent through the address path and converted into the required address to access the

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

local memory for TCD*n*. Next, the TCD memory is accessed and the required descriptor is read from the local memory and then loaded into the eDMA engine address path's primary or secondary channel execution registers. The TCD memory is 64 bits wide to minimize the time needed to fetch the activated channel descriptor and load it into the address path registers.

The following diagram illustrates the second part of the basic data flow:

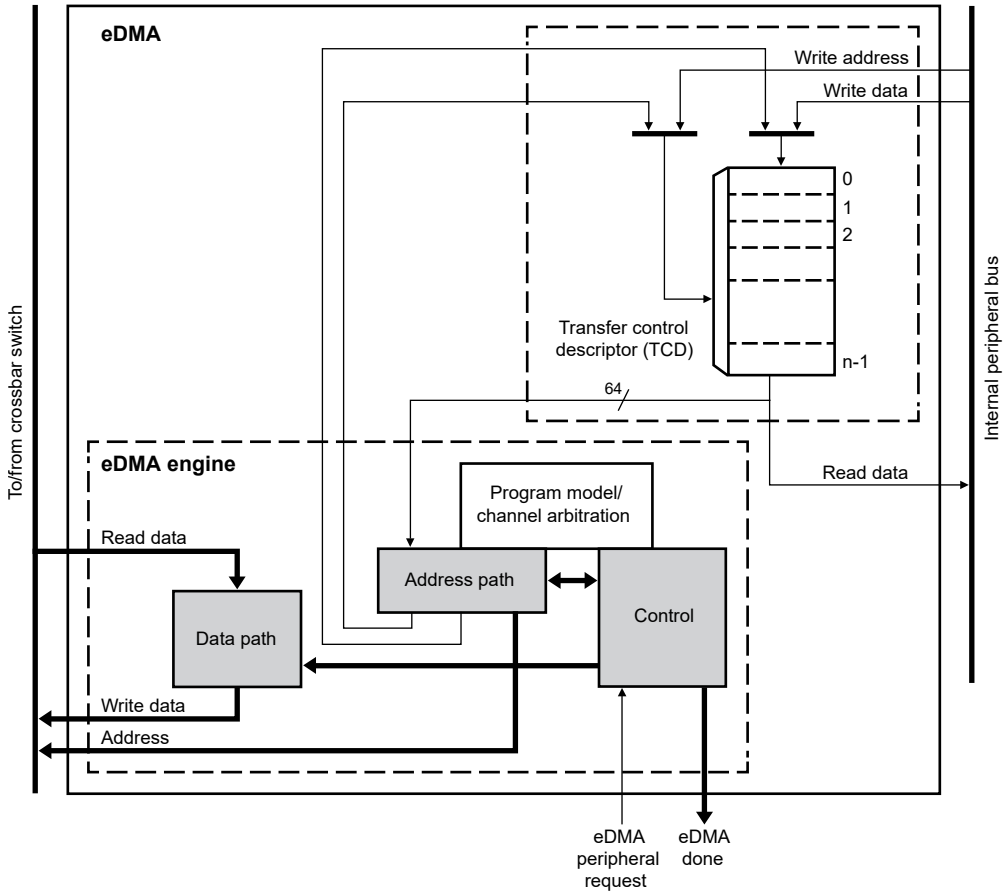


Figure 19. eDMA operation, part 2

The modules associated with the data transfer (address path, data path, and control) go through the required sequence of source reads and destination writes to perform the actual data movement. The source reads are initiated, and the fetched data is temporarily stored in the data path block until it is gated onto the internal bus during the destination write. This source read/destination write processing continues until the byte count, NBYTES, has been transferred.

After NBYTES of data has been moved, the final phase of the basic data flow is performed. In this segment, the address path logic performs the required updates to certain fields in the appropriate TCD (for example, SADDR, DADDR, CITER). If the major iteration count is exhausted, additional operations are performed. These include the final address adjustments and reloading of the BITER field into the CITER field. Assertion of an optional interrupt request also occurs at this time, as does a possible fetch of a new TCD from memory using the scatter/gather address pointer included in the descriptor (if scatter/gather is enabled). The updates to the TCD memory and the assertion of an interrupt request are shown in the following diagram.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

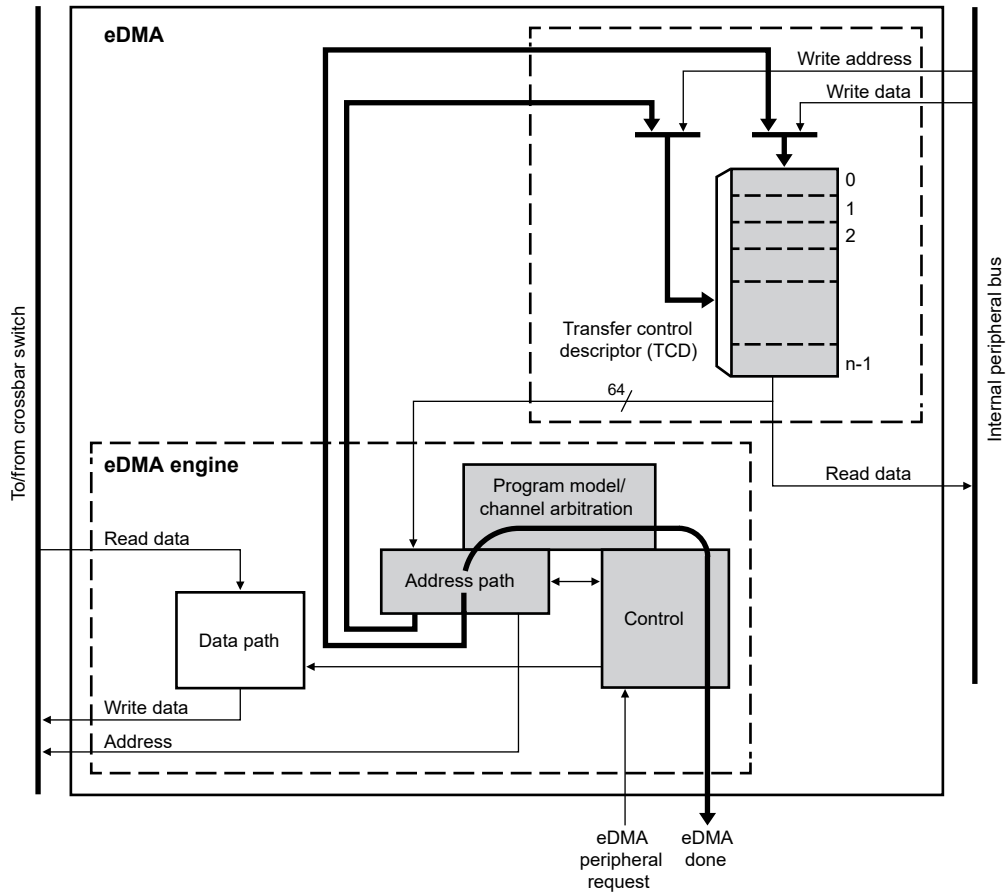


Figure 20. eDMA operation, part 3

14.3.3 Fault reporting and handling

Channel errors are reported in the Error Status register (CHn_ES) and can be caused by any of the following:

- A configuration error, which is an illegal setting in the transfer control descriptor
- An active channel canceled via a "cancel transfer with error" hardware or software request
- An error termination to a bus initiator read or write cycle

A configuration error is reported when an inconsistent state is represented by one of these factors:

- Starting source or destination address
- Source or destination offsets
- Minor loop byte count
- Transfer size

Each of these possible causes is detailed below:

- The addresses and offsets must be aligned on zero-modulo-transfer-sized boundaries.
- The minor loop byte count must be a multiple of the source and destination transfer sizes.
- All source reads and destination writes must be configured to the natural boundary of the programmed transfer size.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

NOTE

To aid in debugging, set the Halt After Error field in the DMA's Control Status register, CSR[HAE]. Upon any error condition, the DMA is halted after the error is recorded. The DMA remains halted and does not process any channel service requests. After the error is fixed, the DMA may be enabled again by clearing the Halt field, CSR[HALT].

- If a scatter/gather operation is enabled upon channel completion, a configuration error is reported if the scatter/gather address (TCDn_DLAST_SGA) is not aligned on a 32-byte boundary.
- If minor loop channel linking is enabled upon channel completion, a configuration error is reported when the link is attempted if the TCDn_CITER[ELINK] field does not equal the TCDn_BITER[ELINK] field.

If enabled, all configuration error conditions, except the scatter/gather and minor-loop link errors, are reported as the channel activates and asserts an error interrupt request. A scatter/gather configuration error is reported when the scatter/gather operation begins at major loop completion if properly enabled. A minor loop channel link configuration error is reported when the link operation is serviced at minor loop completion.

If a system bus read or write is terminated with an error, the data transfer is stopped and the appropriate bus error flag set. In this case, the state of the channel's transfer control descriptor is updated by the eDMA engine with the current source address, destination address, and current iteration count at the point of the fault. When a system bus error occurs, the channel terminates after the next transfer. Due to pipeline effect, the next transfer is already in progress when the bus error is received by the eDMA. If a bus error occurs on the last read prior to beginning the write sequence, the write executes using the data captured during the bus error. If a bus error occurs on the last write prior to switching to the next read sequence, the read sequence executes before the channel terminates due to the destination bus error.

The occurrence of any error causes the eDMA engine to stop normal processing of the active channel immediately (it goes to its error processing states and the transaction to the system bus still has pipeline effect), and the appropriate channel field in the eDMA error register is set to 1. At the same time, the details of the error condition are loaded into the Error Status register (CHn_ES). The major loop complete indicators, setting the transfer control descriptor DONE flag, and the possible assertion of an interrupt request are not affected when an error is detected.

After the error status has been updated, the eDMA engine continues operating by servicing the next appropriate channel. A channel that experiences an error condition is not automatically disabled. If a channel is terminated by an error and then issues another service request before the error is fixed, that channel executes and terminates with the same error condition.

The error status fields are read-only. These error indicators are sticky and cannot be cleared. They show the last recorded error until the DMA is reset. CHn_ES[ERR] is used to determine if a new error condition exists. This field is the logical OR of each channel's error interrupt field (ERR).

After the software has resolved all errors and cleared all of the error interrupt fields, the MP_ES[VLD] is cleared to 0 but the cause of the last error is still indicated.

14.3.4 Channel preemption

The eDMA uses a priority vector value to determine the highest priority channel requesting service.

The priority vector is a combination of:

1. the channel's group priority, CHn_GRPRI
2. the channel's priority level, CHn_PRI[APL]

It can be considered a number composed of these concatenated priority levels: CHn_GRPRI : CHn_PRI[APL]

Priority vector = ((CHn_GRPRI << 8) + (CHn_PRI[APL] << 5) + CH_NUM)

A channel requesting service with the highest priority vector value will receive the next execution slot.

An execution slot is available:

1. immediately if the eDMA is idle
2. when an active channel retires
3. when valid preemption conditions exist

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

NOTE

Preemption is strictly priority based. Preemption is not bound by a specific group number as defined by CHn_GRPRI.

Channel preemption is enabled on a per-channel basis by setting the CHn_PRI[ECP] field. Channel preemption allows the executing channel's data transfers to temporarily suspend in favor of starting a higher-priority channel. After the preempting channel has completed all of its minor loop data transfers, the preempted channel is restored and resumes execution.

After the restored channel completes one read/write sequence, it is again eligible for preemption. If any higher priority channel is requesting service, the restored channel is suspended, and the higher-priority channel is serviced. Nested preemption, that is, attempting to preempt a preempting channel, is not supported. After a preempting channel begins execution, it cannot be preempted.

A channel's ability to preempt another channel can be disabled by setting CHn_PRI[DPA] to 1. When a channel's preempt ability is disabled, that channel cannot suspend a lower-priority channel's data transfer, regardless of the lower-priority channel's ECP setting. This allows for a pool of low-priority, large-data-moving channels to be defined.

You can configure these low-priority channels to not preempt each other, thus preventing a low-priority channel from consuming the preempt slot normally available to a true high-priority channel. When you enable round-robin channel arbitration mode (CSR[ERCA] is set to 1), any channel with a priority level equal to 0 (CHn_PRI[APL] = 0) has preemption disabled and cannot preempt another channel.

14.3.5 Clocking

The following table describes the clock sources:

Table 81. Clocks

Clock	Description
hclk	Clock used for the DMA engine
ipg_clk	Clock used for the DMA programming model
dma_ram_clk	Clock used for the TCD RAMs

14.3.6 Interrupts

Software can enable the interrupt for each channel for the following events:

1. The major loop is half complete (INTHALF)
2. The major loop is complete (INTMAJOR)
3. A configuration error occurs (EEI)

14.4 External signals

This module has no external signals.

14.5 Initialization

The following sections discuss initialization of the eDMA and programming considerations.

14.5.1 eDMA initialization

To initialize the eDMA:

1. Write to the MP_CSR if a configuration other than the default is wanted.
2. Write the channel priority levels to the CHn_PRI registers and group priority levels to the CHn_GRPRI registers if a configuration other than the default is wanted.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

3. Enable error interrupts in the CHn_CSR[EEI] registers if they are wanted.
4. Write the 32-byte TCD for each channel that may request service.
5. Enable any hardware service requests via the CHn_CSR[ERQ] registers.
6. Request channel service via either:
   - Software: setting TCDn_CSR[START]
   - Hardware: target device asserting its eDMA peripheral request signal

After any channel requests service, a channel is selected for execution based on the arbitration and priority levels written into the programmer's model. The eDMA engine reads the entire TCD, including the TCD control and status fields, as shown in Table 82, for the selected channel into its internal address path module.

As the TCD is read, the first transfer is initiated on the internal bus, unless a configuration error is detected. Transfers from the source, defined by TCDn_SADDR, to the destination, defined by TCD_DADDR, continue until the number of bytes specified by TCDn_NBYTES are transferred.

When the transfer is complete, the eDMA engine's local TCDn_SADDR, TCDn_DADDR, and TCDn_CITER are written back to the main TCD memory and any minor loop channel linking is performed, if enabled. If the major loop is exhausted, then eDMA executes further post-processing, such as interrupts, major loop channel linking, and scatter/gather operations, if enabled.

Table 82. TCD control and status (TCDn_CSR) fields

TCDn_CSR field name	Description
START	Control field to start the channel explicitly when using a software-initiated DMA service (automatically cleared by hardware)
EEOP	Control field to enable end-of-packet processing
ESDA	Control field to enable storing of the destination address to system memory after the major loop completes
DREQ	Control field to disable hardware-initiated DMA service requests after major loop completion
BWC	Control field for throttling the bandwidth control of a channel
ESG	Control field to enable the scatter-gather feature
INTHALF	Control field to enable interrupt when major loop is half-complete
INTMAJOR	Control field to enable interrupt when major loop completes

Table 83. Channel control and status (CHn_CSR) fields

CHn_CSR field name	Description
ACTIVE	Status field indicating the channel is currently in execution
DONE	Status field indicating major loop completion (cleared by software when a channel begins execution)
EEI	Control field to enable error interrupts
EARQ	Control field to enable external, asynchronous wakeup event in conjunction with the ERQ field
ERQ	Control field to enable hardware service requests

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

The following figure shows how each DMA request initiates one minor-loop transfer, or iteration, without CPU intervention. DMA arbitration can occur after each minor loop, and one level of minor loop DMA preemption is allowed. The number of minor loops in a major loop is specified by the beginning iteration count (BITER).

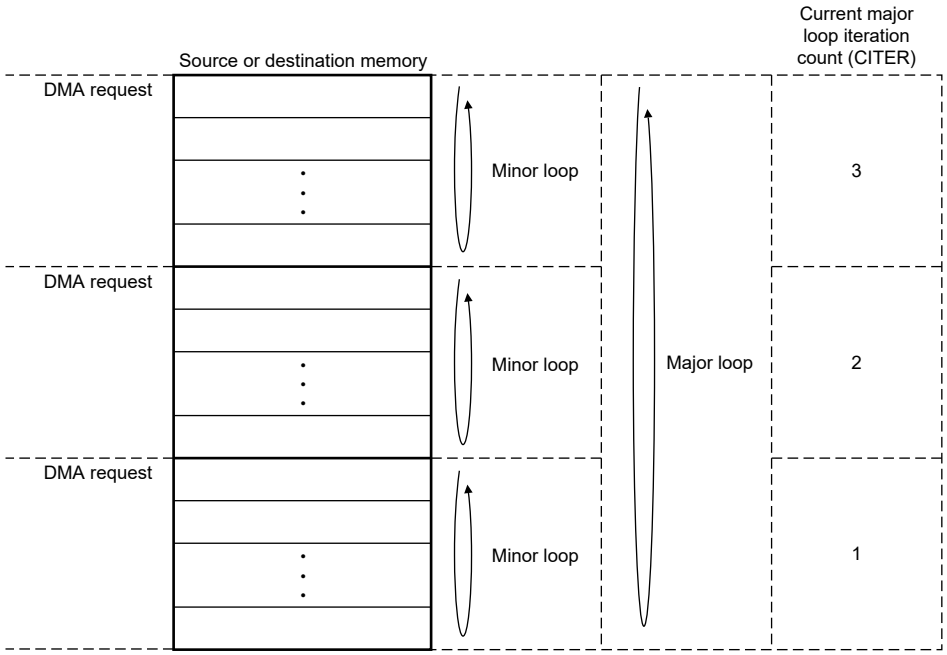


Figure 21. Example of multiple loop iterations

The following figure lists the memory array terms and how the TCD settings are related.

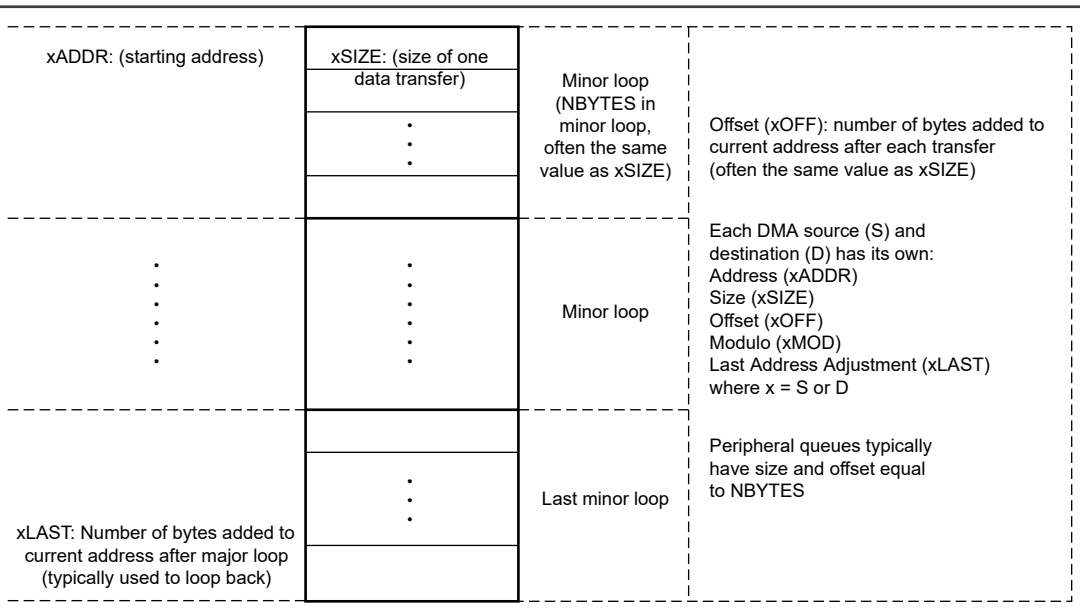


Figure 22. Memory array terms

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

14.5.2 eDMA arbitration

The eDMA uses a layered arbitration scheme composed of multiple priority levels. The eDMA uses a fixed-priority arbitration scheme with optional round-robin arbitration under specific conditions. The priorities are evaluated in the following order:

Table 84. eDMA arbitration priorities

Priority	Scheme	Description
1 (Highest)	Arbitration group priority	Each channel is assigned an arbitration group via the CHn_GRPRI registers. Priority is given to the highest value (31 being the highest possible value) down to the lowest value (zero, the default).
2	Channel priority	Each channel is assigned a channel priority level via the CHn_PRI registers. The channel priority is a relative priority level within an arbitration group. Priority is given to the highest value (seven being the highest possible value) down to the lowest value (zero, the default). Channel priorities within each arbitration group need not be unique. If multiple channels have the same channel priority level, the channel number will be used to determine priority as defined in row three.
3	Channel number	When two or more channels have the same arbitration group priority and channel priority, the channel number (CHn_NUM) is used to determine the highest priority. Priority is given to the highest channel number. Lowest priority is channel 0. The channel numbers are static and cannot be changed in the programmer's model.
4 (Lowest)	Round-robin	When round-robin is enabled, any channel configured for round-robin operation has lowest priority within an arbitration group. Round-robin is enabled by setting the MP_CSR[ERCA] field to 1. After being enabled, channels with a channel priority of zero (CHn_PRI=0) will use round-robin arbitration. Round-robin arbitration will rotate the channel selection among the channels requesting service with CHn_PRI=0 within the arbitration group. Any non-zero channel within the arbitration group will continue to use fixed-priority arbitration, and if requesting service will be selected over any round-robin channels.

For fixed arbitration, the overall priority can be considered a number composed of three concatenated priority levels: CHn_GRPRI:CHn_PRI:CH_NUM. The largest number has the highest priority and the lowest number has the lowest priority.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

For round-robin arbitration, the priority number is CHn_GRPRI:0:X. The module rotates through the CHn_PRI=0 channels requesting service without regard to priority among these channels. Any channel within the arbitration group for which CHn_PRI is greater than 0 will be serviced before the round-robin channels.

14.5.3 Programming errors

The eDMA performs various tests on the transfer control descriptor to verify consistency in the descriptor data.

The channel number causing the error is recorded in the Error Status register (CHn_ES). If the error source is not removed before the next activation of the problematic channel, the error is detected and recorded again. Setting the halt after error field, CSR[HAE], will halt the DMA and prevent recurrence of the error.

14.5.4 Arbitration mode considerations

This section discusses arbitration considerations for eDMA.

14.5.4.1 Fixed group arbitration, fixed channel arbitration

In this mode, eDMA selects for execution the channel service request from the highest-priority channel in the highest-priority group. If eDMA is programmed so that the channels within a high-priority group have a high number of requests or large data transfers, that group may consume all the bandwidth of the eDMA controller. That is, no lower-priority groups are serviced if there is always at least one DMA request pending on a channel in the highest-priority group when the controller arbitrates the next DMA request. The advantage of this scenario is that latency can be small for channels that need to be serviced quickly.

14.5.4.2 Fixed group arbitration, round-robin channel arbitration

The highest-priority group with a request is serviced. Lower-priority groups are serviced if no pending requests exist in the higher-priority groups.

Within each group, channels are serviced starting with the highest non-zero channel priority. For all channels with a channel priority programmed to 0, selection begins with the highest channel number requesting service and then rotates through to the lowest channel number requesting service. The round-robin channel arbitration can provide a fairness mechanism to lower-priority channels.

This scenario could cause the same bandwidth consumption problem as indicated in Fixed group arbitration, fixed channel arbitration, but all the channels in the highest-priority group will be serviced. Service latency is short on the highest-priority group, but could potentially be very much longer as the group priority decreases.

14.5.5 Performing DMA transfers

This section presents examples on how to perform DMA transfers with the eDMA.

14.5.5.1 Single request

To perform a simple transfer of n bytes of data with one activation, set the major loop to one (TCD*n*_CITER = TCD*n*_BITER = 1). The data transfer begins after the channel service request is acknowledged and the channel is selected to execute. After the transfer is complete, the CH*n*_CSR[DONE] field is set to 1 and an interrupt is generated if properly enabled.

For example, the following TCD entry is configured to transfer 16 bytes of data. The eDMA is programmed for one iteration of the major loop transferring 16 bytes per iteration. The source memory has a byte-wide memory port located at 0x1000. The destination memory has a 32-bit port located at 0x2000. The address offsets are programmed in increments to match the transfer size: one byte for the source, and four bytes for the destination. The final source and destination addresses are adjusted to return to their beginning values.

```
TCDn_CITER = TCDn_BITER = 1
TCDn_NBYTES = 16
TCDn_SADDR = 0x1000
TCDn_SOFF = 1
TCDn_ATTR[SSIZE] = 0
```


All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

```
TCDn_SLAST = -16
TCDn_DADDR = 0x2000
TCDn_DOFF = 4
TCDn_ATTR[DSIZE] = 2
TCDn_DLAST_SGA= -16
TCDn_CSR[INTMAJ] = 1
TCDn_CSR[START] = 1 (should be written last after all other fields have been initialized)
All other TCDn fields = 0
```

This generates the following event sequence:

1. User write to the TCD*n*_CSR[START] field requests channel service.
2. The channel is selected by arbitration for servicing.
3. eDMA engine writes:
   - CH*n*_CSR[DONE] = 0
   - TCD*n*_CSR[START] = 0
   - CH*n*_CSR[ACTIVE] = 1
4. eDMA engine reads: channel TCD data from local memory to internal register file.
5. The source-to-destination transfers are executed as follows:
   a. Read byte from location 0x1000, read byte from location 0x1001, read byte from 0x1002, read byte from 0x1003.
   b. Write 32 bits to location 0x2000 → first iteration of the minor loop.
   c. Read byte from location 0x1004, read byte from location 0x1005, read byte from 0x1006, read byte from 0x1007.
   d. Write 32 bits to location 0x2004 → second iteration of the minor loop.
   e. Read byte from location 0x1008, read byte from location 0x1009, read byte from 0x100A, read byte from 0x100B.
   f. Write 32 bits to location 0x2008 → third iteration of the minor loop.
   g. Read byte from location 0x100C, read byte from location 0x100D, read byte from 0x100E, read byte from 0x100F.
   h. Write 32 bits to location 0x200C → last iteration of the minor loop → major loop complete.
6. The eDMA engine writes: TCD*n*_SADDR = 0x1000, TCD*n*_DADDR = 0x2000, TCD*n*_CITER = 1 (TCD*n*_BITER).
7. The eDMA engine writes: CH*n*_CSR[ACTIVE] = 0, CH*n*_CSR[DONE] = 1, CH*n*_INT[INT] = 1.
8. The channel retires and the eDMA goes idle or services the next channel.

14.5.5.2 Multiple requests

The following example transfers 32 bytes via two hardware requests, but is otherwise the same as the previous example. The only fields that change are the major loop iteration count and the final address offsets. The eDMA is programmed for two iterations of the major loop, transferring 16 bytes per iteration. After the channel's hardware requests are enabled via the CHn_CSR[ERQ] register field, the target device initiates channel service requests.

```
TCDn_CITER = TCDn_BITER = 2
TCDn_SLAST = -32
TCDn_DLAST_SGA = -32
```

This would generate the following sequence of events:

1. First hardware (eDMA peripheral) requests channel service.
2. The channel is selected by arbitration for servicing.
3. eDMA engine writes: CH*n*_CSR[DONE] = 0, TCD*n*_CSR[START] = 0, CH*n*_CSR[ACTIVE] = 1.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

4. eDMA engine reads: channel TCD*n* data from local memory to internal register file.
5. The source-to-destination transfers are executed as follows:
    a. Read byte from location 0x1000, read byte from location 0x1001, read byte from 0x1002, read byte from 0x1003.
    b. Write 32 bits to location 0x2000 → first iteration of the minor loop.
    c. Read byte from location 0x1004, read byte from location 0x1005, read byte from 0x1006, read byte from 0x1007.
    d. Write 32 bits to location 0x2004 → second iteration of the minor loop.
    e. Read byte from location 0x1008, read byte from location 0x1009, read byte from 0x100A, read byte from 0x100B.
    f. Write 32 bits to location 0x2008 → third iteration of the minor loop.
    g. Read byte from location 0x100C, read byte from location 0x100D, read byte from 0x100E, read byte from 0x100F.
    h. Write 32 bits to location 0x200C → last iteration of the minor loop.
6. eDMA engine writes: TCD*n*_SADDR = 0x1010, TCD*n*_DADDR = 0x2010, TCD*n*_CITER = 1.
7. eDMA engine writes: CH*n*_CSR[ACTIVE] = 0.
8. The channel retires, which concludes one iteration of the major loop. The eDMA goes idle or services the next channel.
9. Second hardware (eDMA peripheral) requests channel service.
10. The channel is selected by arbitration for servicing.
11. eDMA engine writes: CH*n*_CSR[DONE] = 0, TCD*n*_CSR[START] = 0, CH*n*_CSR[ACTIVE] = 1.
12. eDMA engine reads: Channel TCD data from local memory to internal register file.
13. The source-to-destination transfers are executed as follows:
    a. Read byte from location 0x1010, read byte from location 0x1011, read byte from 0x1012, read byte from 0x1013.
    b. Write 32 bits to location 0x2010 → first iteration of the minor loop.
    c. Read byte from location 0x1014, read byte from location 0x1015, read byte from 0x1016, read byte from 0x1017.
    d. Write 32 bits to location 0x2014 → second iteration of the minor loop.
    e. Read byte from location 0x1018, read byte from location 0x1019, read byte from 0x101A, read byte from 0x101B.
    f. Write 32 bits to location 0x2018 → third iteration of the minor loop.
    g. Read byte from location 0x101C, read byte from location 0x101D, read byte from 0x101E, read byte from 0x101F.
    h. Write 32 bits to location 0x201C → last iteration of the minor loop → major loop complete.
14. eDMA engine writes: TCD*n*_SADDR = 0x1000, TCD*n*_DADDR = 0x2000, TCD*n*_CITER = 2 (TCD*n*_BITER).
15. eDMA engine writes: CH*n*_CSR[ACTIVE] = 0, CH*n*_CSR[DONE] = 1, CH*n*_INT[INT] = 1.
16. The channel retires, which concludes with the major loop complete. The eDMA goes idle or services the next channel.

14.5.5.3 Using the modulo feature

The modulo feature of the eDMA allows implementation of a circular data queue in which the size of the queue is a power of 2. xMOD is a 5-bit field for the source and destination in the TCD, and it specifies which lower address bits increment from their original value after the address+offset calculation. All upper address bits remain the same as in the original value. A setting of 0 for this field disables the modulo feature. Modulo addressing applies to cases where the minor loop offset is enabled; that is, the upper address bits remain the same after the minor loop offset is added to the source or destination address.

The following table shows how the transfer addresses are specified based on the setting of the MOD field. Here a circular buffer is created where the address wraps to the original value but the 28 upper address bits (0x1234567*x*) retain their original value. In this example, the source address is set to 0x12345670, the offset is set to four bytes, and the MOD field is set to four, which allows for a 2^4 byte (16 byte) queue size.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table 85. Modulo example

Transfer number	Address
1	0x12345670
2	0x12345674
3	0x12345678
4	0x1234567C
5	0x12345670
6	0x12345674

14.5.6 Monitoring transfer descriptor status

This section discusses how to monitor eDMA status.

14.5.6.1 Testing for minor loop completion

There are two methods to test for minor loop completion when using software-initiated service requests.

1. The first method is to read the TCDn_CITER field and test for a change.
2. The second method, extracted from the sequence shown below, is to test the TCDn_CSR[START] field and the CHn_CSR[ACTIVE] field. The minor-loop-complete condition is indicated by both fields reading 0 after TCDn_CSR[START] is set to 1. Polling the CHn_CSR[ACTIVE] field only may be inconclusive because the active status may be missed if the channel execution is short in duration.

The CHn_CSR and TCDn_CSR status fields execute the following sequence for a software-activated channel:

Stage	TCDn_CSR field	CHn_CSR fields		State
	START	ACTIVE	DONE	
1	1	0	0	Initiate channel service request via software.
2	0	1	0	Channel is executing.
3a	0	0	0	Channel has completed the minor loop and is idle.
3b	0	0	1	Channel has completed the major loop and is idle.

The best method to test for minor-loop completion when using hardware-initiated (that is, peripheral-initiated) service requests is to read the TCDn_CITER field and test for a change. The hardware request and acknowledge handshake signals are not visible in the programmer's model.

The TCD status fields execute the following sequence for a hardware-activated channel:

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Stage	TCDn_CSR field	CHn_CSR fields		State
	START	ACTIVE	DONE	
1	0	0	0	Initiate channel service request via hardware (peripheral request asserted).
2	0	1	0	Channel is executing.
3a	0	0	0	Channel has completed the minor loop and is idle.
3b	0	0	1	Channel has completed the major loop and is idle.

For both activation types, the major-loop-complete status is explicitly indicated via the CHn_CSR[DONE] field.

The TCDn_CSR[START] field is cleared to 0 automatically when the channel begins execution, regardless of how the channel activates.

14.5.6.2 Reading the transfer descriptors of active channels

The eDMA reads back the true TCDn_SADDR, TCDn_DADDR, and TCDn_NBYTES values if they are read when a channel executes. The true values of SADDR, DADDR, and NBYTES are the values the eDMA engine currently uses in its internal register file, and not the values in the TCD local memory for that channel. The addresses, SADDR and DADDR, and NBYTES (which decrements to zero as the transfer progresses), can give an indication of the progress of the transfer. All other values are read back from the TCD local memory.

14.5.6.3 Checking channel preemption status

A preemptive situation is one in which a preempt-enabled channel is executing and a higher-priority request becomes active. When round-robin channel arbitration mode is enabled, all channels with their channel priority set to 0 lose their preempt ability. Channel priorities of 0 are treated as equal, that is, they are constantly rotating, when round-robin arbitration mode is enabled.

The CHn_CSR[ACTIVE] field for the preempted channel remains asserted throughout the preemption. The preempted channel is temporarily suspended when the preempting channel executes one major loop iteration. If two CHn_CSR[ACTIVE] fields are set simultaneously in the global TCD map, a higher-priority channel is actively preempting a lower-priority channel.

14.5.7 Channel linking

Channel linking (or chaining) is a mechanism in which one channel sets the TCDn_CSR[START] field of another channel (or itself), thus initiating a service request for that channel. When properly enabled, the eDMA engine automatically performs this operation at the major or minor loop completion.

The minor loop channel linking occurs at the completion of the minor loop (or one iteration of the major loop). The TCDn_CITER[ELINK] field determines whether a minor loop link is requested. When enabled, the channel link is made after each iteration of the major loop except for the last. When the major loop is exhausted, only the major loop channel link fields are used to determine if a channel link should be made. For example, using an initial field setting of:

```
MP_CSR[GCLC] = 1
TCDn_CITER[ELINK] = 1
TCDn_CITER[LINKCH] = 0xC
TCDn_CITER[CITER] value = 0x4
TCDn_CSR[MAJORELINK] = 1
TCDn_CSR[MAJORLINKCH] = 0x7
```

executes as:

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

1. Minor loop done → set TCD12_CSR[START] field
2. Minor loop done → set TCD12_CSR[START] field
3. Minor loop done → set TCD12_CSR[START] field
4. Minor loop done, major loop done→ set TCD7_CSR[START] field

When minor loop linking is enabled (TCDn_CITER[ELINK] = 1), the TCDn_CITER[CITER] field uses a nine-bit vector to form the current iteration count. When minor loop linking is disabled (TCDn_CITER[ELINK] = 0), the TCDn_CITER[CITER] field uses a 15-bit vector to form the current iteration count. The bits associated with the TCDn_CITER[LINKCH] field are concatenated onto the CITER value to increase the range of the CITER.

NOTE

The TCDn_CITER[ELINK] field and the TCDn_BITER[ELINK] field must be equal — if they are not, a configuration error is reported. The CITER and BITER vector widths must be equal to calculate the major loop halfway done interrupt point.

The following table summarizes how a DMA channel can link to another DMA channel, that is, use another channel's TCD, at the end of a loop.

Table 86. Channel linking parameters

Wanted link behavior	TCD control field name	Description
Link at end of minor loop	TCDn_CITER[ELINK]	Enable channel-to-channel linking on minor loop completion (current iteration)
	TCDn_CITER[LINKCH]	Link channel number when linking at end of minor loop (current iteration)
Link at end of major loop	TCDn_CSR[MAJORELINK]	Enable channel-to-channel linking on major loop completion
	TCDn_CSR[MAJORLINKCH]	Link channel number when linking at end of major loop

14.5.8 Dynamic programming

This section provides recommended methods to change the programming model during channel execution.

Steps for channel linking:

1. Set GCLC = 1
2. For linking after major loop complete:
   a. Set MAJORLINKCH = target channel number
   b. Set MAJORELINK = 1, enable the major loop link
   c. Upon major loop completion (CITER decrements to 0), the link will be made.
3. For linking on minor loop completion:
   a. Set LINKCH = target channel number
   b. Set ELINK = 1, enable the minor loop link
   c. Set the beginning iteration count to the same values; LINKCH = target channel and ELINK = 1,
   d. Upon minor loop completion (that is, after each service request except when the major loop is complete) the link will be made.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

14.5.8.1 Dynamically changing the channel priority

To change group or channel priority levels:

1. Halt the DMA by writing 1 to the CSR[HALT] field.
2. Change the group or channel priorities as wanted.
3. Enable normal DMA operations by writing 0 to the CSR[HALT] field.

14.5.8.2 Dynamic channel linking

Dynamic channel linking is the process of setting the TCDn_CSR[MAJORELINK] field during channel execution (see the diagram in TCD structure). This field is read from the TCD local memory at the end of channel execution, thus allowing you to enable the feature during channel execution.

Because you are allowed to change the configuration during execution, you need a coherency model. Consider the scenario where you attempt to execute a dynamic channel link by enabling the TCDn_CSR[MAJORELINK] field at the same time the eDMA engine is retiring the channel. TCDn_CSR[MAJORELINK] would be set in the programmer's model, but it would be unclear whether the actual link was made before the channel retired.

We recommend that you use the following coherency model when executing a dynamic channel link request.

1. Write 1 to the TCDn_CSR[MAJORELINK] field.
2. Read back the TCDn_CSR[MAJORELINK] field.
3. Test the TCDn_CSR[MAJORELINK] request status:
   - If TCDn_CSR[MAJORELINK] = 1, the dynamic link attempt was successful.
   - If TCDn_CSR[MAJORELINK] = 0, the attempted dynamic link did not succeed (the channel was already retiring).

For this request, the TCD local memory controller forces the TCDn_CSR[MAJORELINK] field to 0 on any writes to a channel's TCDn_CSR[7:0] after that channel's CHn_CSR[DONE] field is set to 1, indicating the major loop is complete.

NOTE

You must clear the CHn_CSR[DONE] field to 0 before writing to the TCDn_CSR[MAJORELINK] field. The CHn_CSR[DONE] field is cleared to 0 automatically by the eDMA engine after a channel begins execution.

14.5.8.3 Dynamic scatter/gather

Scatter/gather is the process of automatically loading a new TCD into a channel. It allows a DMA channel to use multiple TCDs; this enables a DMA channel to scatter the DMA data to multiple destinations or gather it from multiple sources. When scatter/gather is enabled and the channel has finished its major loop, a new TCD is fetched from system memory and loaded into that channel's descriptor location in the eDMA programmer's model, thus replacing the current descriptor.

Because you are allowed to change the configuration during execution, you need a coherency model. Consider the scenario where you attempt to execute a dynamic scatter/gather operation by enabling the TCDn_CSR[ESG] field at the same time the eDMA engine is retiring the channel. The TCDn_CSR[ESG] field would be set in the programmer's model, but it would be unclear whether the actual scatter/gather request was honored before the channel retired.

Two methods are recommended for executing a dynamic scatter/gather request. Whenever the TCDn_CSR is written, the TCD local memory controller forces the TCDn_CSR[ESG] field to 0 on any writes to a channel's TCDn_CSR[7:0] after that channel's CHn_CSR[DONE] field has been set to 1, indicating the major loop is complete. If attempting to set the ESG, ensure the DONE field is cleared to 0.

NOTE

You must clear the CHn_CSR[DONE] field to 0 before writing the TCDn_CSR[MAJORELINK] or TCDn_CSR[ESG] fields. The CHn_CSR[DONE] field is cleared to 0 automatically by the eDMA engine after a channel begins execution and is set to 1 upon major loop completion.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

14.5.8.3.1 Method 1 (channel not using major loop channel linking)

For a channel not using major loop channel linking, the coherency model described here may be used for a dynamic scatter/gather request.

When the TCDn_CSR[MAJORELINK] field is 0, the TCDn_CSR[MAJORLINKCH] field is not used by the eDMA. In this case, the TCDn_CSR[MAJORLINKCH] bits may be used for other purposes. This method uses the TCDn_CSR[MAJORLINKCH] field as a TCDn_CSR identification (ID).

When the descriptors are built, write a unique TCDn_CSR ID in the TCDn_CSR[MAJORLINKCH] field for each TCDn_CSR associated with a channel using dynamic scatter/gather.

1. Write a 1 to the TCDn_CSR[DREQ] field. Should a dynamic scatter/gather attempt fail, setting the TCDn_CSR[DREQ] field to 1 will prevent future hardware activation of this channel. This stops the channel from executing with a destination address (daddr) that was calculated using a scatter/gather address (written in the next step) instead of a DLAST final offset value.
2. Write the TCDn_DLAST_SGA field with the scatter/gather address.
3. Write a 1 to the TCDn_CSR[ESG] field.
4. Read back the 16-bit TCDn_CSR control/status field.
5. Test the TCDn_CSR[ESG] request status and TCDn_CSR[MAJORLINKCH] value:
   - If ESG = 1, the dynamic scatter/gather attempt was successful.
   - If ESG = 0 and the MAJORLINKCH (ID) did not change, the dynamic scatter/gather attempt was not successful (the channel was already retiring).
   - If ESG = 0 and the MAJORLINKCH (ID) changed, the dynamic scatter/gather attempt was successful (the new TCDn_CSR's ESG value cleared the ESG field to 0).

14.5.8.3.2 Method 2 (channel using major loop channel linking)

For a channel using major loop channel linking, the coherency model described here may be used for a dynamic scatter/gather request. This method uses the TCDn_DLAST_SGA field as a TCD identification (ID).

1. Write a 1 to the TCDn_CSR[DREQ] field. Should a dynamic scatter/gather attempt fail, setting the DREQ field to 1 will prevent a future hardware activation of this channel. This stops the channel from executing with a destination address (DADDR) that was calculated using a scatter/gather address (written in the next step) instead of a DLAST final offset value.
2. Write the TCDn_DLAST_SGA field with the scatter/gather address.
3. Write a 1 to the TCDn_CSR[ESG] field.
4. Read back the TCDn_CSR[ESG] field.
5. Test the TCDn_CSR[ESG] request status:
   - If ESG = 1, the dynamic scatter/gather attempt was successful.
   - If ESG = 0, read the 32-bit TCDn_DLAST_SGA field.
   - If ESG = 0 and the TCDn_DLAST_SGA did not change, the dynamic scatter/gather attempt was not successful (the channel was already retiring).
   - If ESG = 0 and the TCDn_DLAST_SGA changed, the dynamic scatter/gather attempt was successful (the new TCDn_CSR's ESG value cleared the ESG field to 0).

14.5.9 Suspend/resume a DMA channel with active hardware service requests

The DMA allows you to move data from memory or peripheral registers to another location in memory or to peripheral registers without CPU interaction. After the DMA and peripherals are configured and active, it is rare but supported to suspend a peripheral's

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

service request dynamically. In this scenario, there are certain restrictions to disabling a DMA hardware service request. For coherency, you must follow a specific procedure.

14.5.9.1 Suspend an active DMA channel

To suspend an active DMA channel:

1. Stop the DMA service request at the peripheral first. Confirm it has been disabled by reading back the appropriate register in the peripheral.
2. Check the DMA's Hardware Request Status (MP_HRS) to ensure there is no service request to the DMA channel being suspended. Then disable the hardware service request by clearing the ERQ field to 0 on the appropriate DMA channel.

For example, assume the SPI is set as a initiator for transmitting data via a DMA service request when the TXFIFO has an empty slot. The DMA will transfer the next command and data to the TXFIFO upon the request. If you need to suspend the DMA/SPI transfer loop, perform the following steps:

1. Disable the DMA service request at the SPI.
2. Ensure there is no DMA service request from the SPI by verifying that MP_HRS[HRS] is 0 for the appropriate channel. If no service request is present, disable the DMA channel by clearing the channel's ERQ field to 0. If a service request is present, wait until the request has been processed and the HRS field reads 0.

14.5.9.2 Resume a DMA channel

To resume a DMA channel:

1. Enable the DMA service request on the appropriate channel by setting its ERQ field to 1.
2. Enable the DMA service request at the peripheral.

14.6 Memory map and register definition

The eDMA programming model is partitioned into three parts:

1. The first part defines a number of registers providing overall control functions and is known as the management page.
2. The second part corresponds to the channel (CH) control, status, and configuration.
3. The third part corresponds to the local TCD memory.

TCD memory

Each channel requires a 32-byte transfer control descriptor for defining the data movement operation. Each TCDn definition is presented as 11 registers of 16 or 32 bits. See EDMA_TCD register descriptions for details.

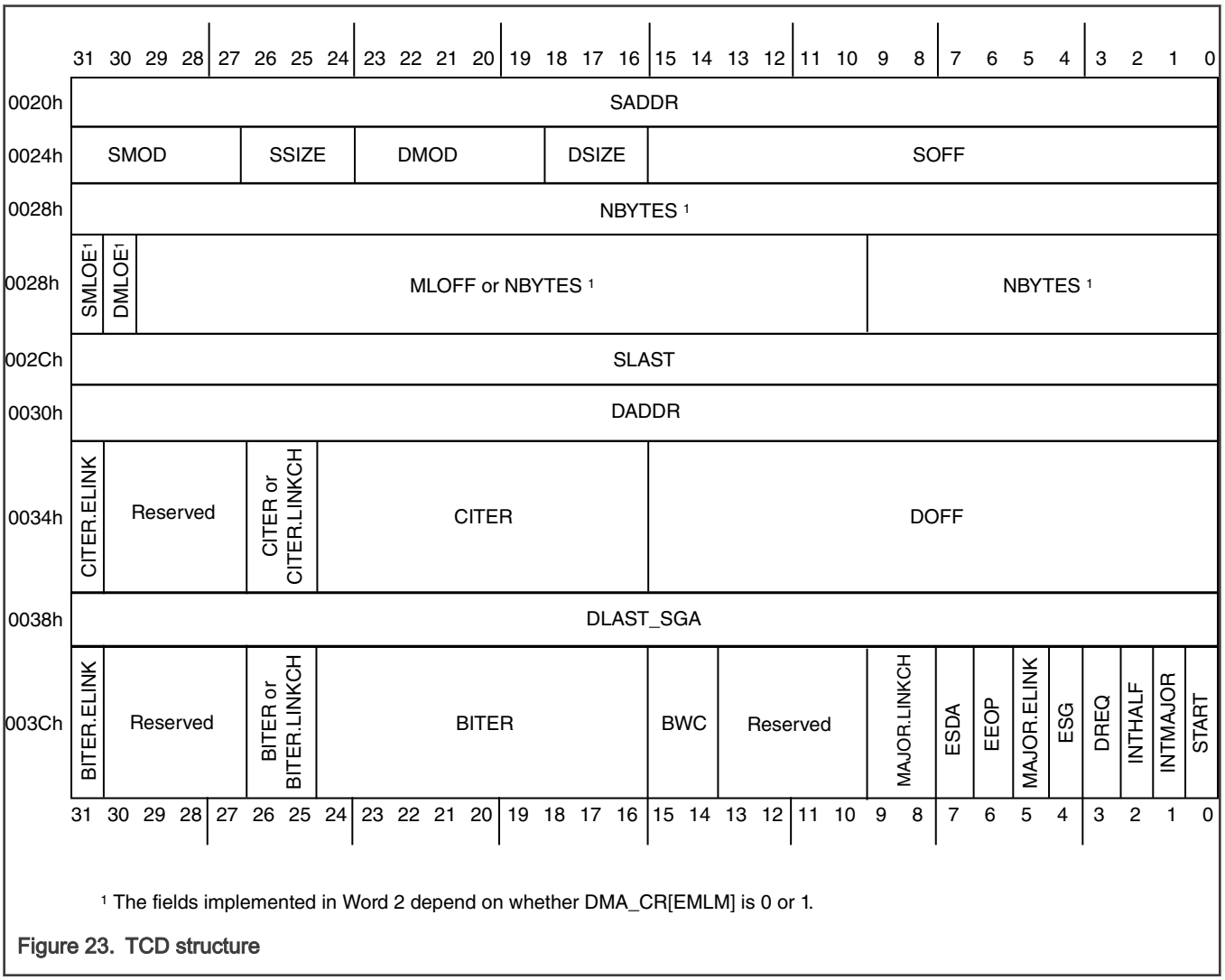
TCD initialization

Prior to activating a channel, you must initialize its TCD with the appropriate transfer profile.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

TCD structure



1 The fields implemented in Word 2 depend on whether DMA_CR[EMLM] is 0 or 1.

Figure 23. TCD structure

Accesses to reserved memory and fields

- Reading reserved fields in a register returns the value of zero.
- Writes to reserved fields in a register are ignored.
- Reading or writing a reserved memory location generates a bus error.

14.6.1 EDMA_MP register descriptions

14.6.1.1 EDMA_MP memory map

eDMA_0_MP base address: 4008_0000h

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Offset	Register	Width (In bits)	Access	Reset value
0h	Management Page Control (MP_CSR)	32	RW	0031_0000h
4h	Management Page Error Status (MP_ES)	32	R	0000_0000h
8h	Management Page Interrupt Request Status (MP_INT)	32	R	0000_0000h
Ch	Management Page Hardware Request Status (MP_HRS)	32	R	0000_0000h
100h - 10Ch	Channel Arbitration Group (CH0_GRPRI - CH3_GRPRI)	32	RW	0000_0000h

14.6.1.2 Management Page Control (MP_CSR)

Offset

Register	Offset
MP_CSR	0h

Function

The Management Page Control register defines the basic operating configuration of the DMA.

Arbitration uses a two-tier priority system; group and channel priority. The eDMA assigns each channel to a priority group. Group arbitration is fixed-priority and cannot be changed. Channel arbitration uses fixed priority and may be configured to use a selective round-robin scheme for specified channels within each priority group. For fixed-priority arbitration, eDMA selects for execution the highest priority channel requesting service in the highest priority arbitration group.

The channel priority registers assign the relative priorities within each arbitration group; see CHn_PRI. All channels with a non-zero CHn_PRI value use fixed-priority arbitration.

When you enable round-robin arbitration, all channels with channel priority set to zero do not have a priority and, of those channels requesting service, are cycled through (from high to low channel number) without regard to priority relative to each other within the same priority group. Any channel with a non-zero CHn_PRI value automatically has a higher priority over the round-robin channels. A channel's priority group is assigned in Channel Arbitration Group (CH0_GRPRI - CH3_GRPRI).

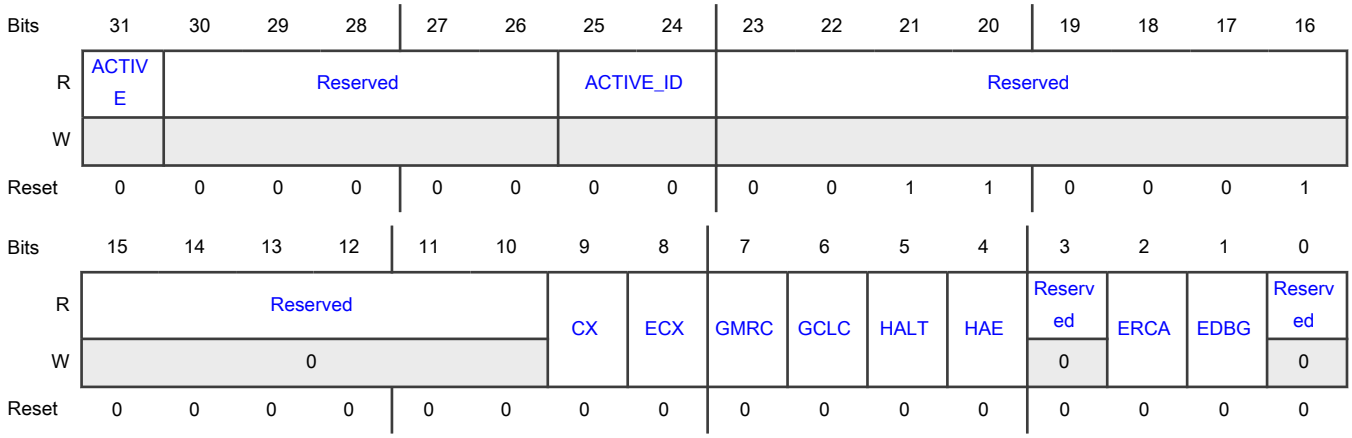
NOTE

For correct operation, changes to the MP_CSR[ERCA, GCLC, GMRC] fields must be performed when the DMA channels are inactive; that is, when the MP_CSR[ACTIVE] field is 0.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram



Fields

Field	Function
31 ACTIVE	DMA Active Status 0b - eDMA is idle 1b - eDMA is executing a channel
30-26 —	Reserved
25-24 ACTIVE_ID	Active Channel ID This field identifies the channel number that is executing when the ACTIVE bit is 1.
23-16 —	Reserved
15-10 —	Reserved
9 CX	Cancel Transfer When set to 1, this field cancels the remaining data transfer, stops the executing channel, and forces the minor loop to finish. The cancel takes effect after the last write of the current read/write sequence. CX clears itself to 0 after the cancel has been honored. This cancel retires the channel normally as if the minor loop had been completed. 0b - Normal operation 1b - Cancel the remaining data transfer
8 ECX	Cancel Transfer With Error Cancellation of the remaining data transfer is similar to that of the CX field. Execution of the channel is stopped and the minor loop is forced to finish. The cancellation takes effect after the last write of the current read/write sequence. The ECX field clears itself to 0 after the cancel is honored. In addition to

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	cancelling the transfer, ECX treats the cancel as an error condition, thus updating Management Page Error Status (MP_ES) and generating an optional error interrupt. 0b - Normal operation 1b - Cancel the remaining data transfer
7 GMRC	Global Initiator ID Replication Control **NOTE** If initiator ID replication is disabled, the privileged protection level (Supervisor mode) for DMA transfers is used. 0b - Initiator ID replication disabled for all channels 1b - Initiator ID replication available and controlled by each channel's CHn_SBR[EMI] setting
6 GCLC	Global Channel Linking Control 0b - Channel linking disabled for all channels 1b - Channel linking available and controlled by each channel's link settings
5 HALT	Halt DMA Operations This field stalls the start of any new channels. Executing channels are allowed to complete. Channel execution resumes when this field is cleared to 0. 0b - Normal operation 1b - Stall the start of any new channels
4 HAE	Halt After Error When this field is set to 1, any error causes the HALT field to be set to 1. Then all service requests are ignored until the HALT field is cleared to 0. 0b - Normal operation 1b - Any error causes the HALT field to be set to 1
3 —	Reserved
2 ERCA	Enable Round Robin Channel Arbitration 0b - Round-robin channel arbitration disabled. Fixed priority arbitration used for channel selection 1b - Round-robin channel arbitration enabled. Round-robin arbitration used for channel selection
1 EDBG	Enable Debug When in debug mode, the DMA stalls the start of a new channel. Executing channels are allowed to complete. DMA resumes channel execution when the system exits debug mode or clears the EDBG field to 0. 0b - Debug mode disabled. When in debug mode, the DMA continues to operate

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	1b - Debug mode is enabled. When in debug mode, the DMA stalls the start of a new channel
0 —	Reserved

14.6.1.3 Management Page Error Status (MP_ES)

Offset

Register	Offset
MP_ES	4h

Function

The ES provides information concerning the last recorded channel error. Channel errors can be caused by:

- An illegal setting in the transfer control descriptor
- An error termination to a bus initiator read or write cycle
- An uncorrectable error that occurred when the device was accessing the TCD SRAM
- A "cancel transfer with error" request was made via the corresponding cancel transfer field or input signal

Upon any error condition, the software must initialize the TCD of the channel that contains the error, as it is in an incomplete state after an error. See Fault reporting and handling for more details.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	VLD	Reserved					ERRCHN		0							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0							ECX	SAE	SOE	DAE	DOE	NCE	SGE	SBE	DBE
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31	Valid Logical OR of all CHn_ES[ERR] status fields.

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
VLD	0b - No CHn_ES[ERR] fields are set to 1 1b - At least one CHn_ES[ERR] field is set to 1, indicating a valid error exists that software has not cleared
30-26 —	Reserved
25-24 ERRCHN	Error Channel Number or Canceled Channel Number The channel number of the last recorded error or last recorded error-canceled transfer.
23-9 —	Reserved
8 ECX	Transfer Canceled The ECX operation is a management page function. When employed, the targeted channel's CHn_ES register reports an unspecified error; that is, only the CHn_ES[ERR] field is set to 1. The management page has full view of the error condition. 0b - No canceled transfers 1b - Last recorded entry was a canceled transfer by the error cancel transfer input
7 SAE	Source Address Error When this field is 1, it indicates that TCDn_SADDR is inconsistent with TCDn_ATTR[SSIZE]. 0b - No source address configuration error 1b - Last recorded error was a configuration error detected in the TCDn_SADDR field
6 SOE	Source Offset Error When this field is 1, it indicates that TCDn_SOFF is inconsistent with TCDn_ATTR[SSIZE]. 0b - No source offset configuration error 1b - Last recorded error was a configuration error detected in the TCDn_SOFF field
5 DAE	Destination Address Error When this field is 1, it indicates that TCDn_DADDR is inconsistent with TCDn_ATTR[DSIZE]. 0b - No destination address configuration error 1b - Last recorded error was a configuration error detected in the TCDn_DADDR field
4 DOE	Destination Offset Error When this field is 1, it indicates that TCDn_DOFF is inconsistent with TCDn_ATTR[DSIZE]. 0b - No destination offset configuration error 1b - Last recorded error was a configuration error detected in the TCDn_DOFF field
3	NBYTES/CITER Configuration Error

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
NCE	This error indicates that one of the following has occurred: • TCDn_NBYTES is not a multiple of TCDn_ATTR[SSIZE] and TCDn_ATTR[DSIZE] • TCDn_CITER[CITER] is equal to zero • TCDn_CITER[ELINK] is not equal to TCDn_BITER[ELINK] 0b - No NBYTES/CITER configuration error 1b - The last recorded error was NBYTES equal to zero or a CITER not equal to BITER error. Last recorded error was a configuration error detected in the TCDn_NBYTES or TCDn_CITER fields
2 SGE	Scatter/Gather Configuration Error When this field is 1, it indicates that TCDn_DLAST_SGA is not on a 32-byte boundary. This field is checked at the beginning of a scatter/gather operation after major loop completion if TCDn_CSR[ESG] is enabled. 0b - No scatter/gather configuration error 1b - Last recorded error was a configuration error detected in the TCDn_DLAST_SGA field
1 SBE	Source Bus Error 0b - No source bus error 1b - Last recorded error was a bus error on a source read
0 DBE	Destination Bus Error 0b - No destination bus error 1b - Last recorded error was a bus error on a destination write

14.6.1.4 Management Page Interrupt Request Status (MP_INT)

Offset

Register	Offset
MP_INT	8h

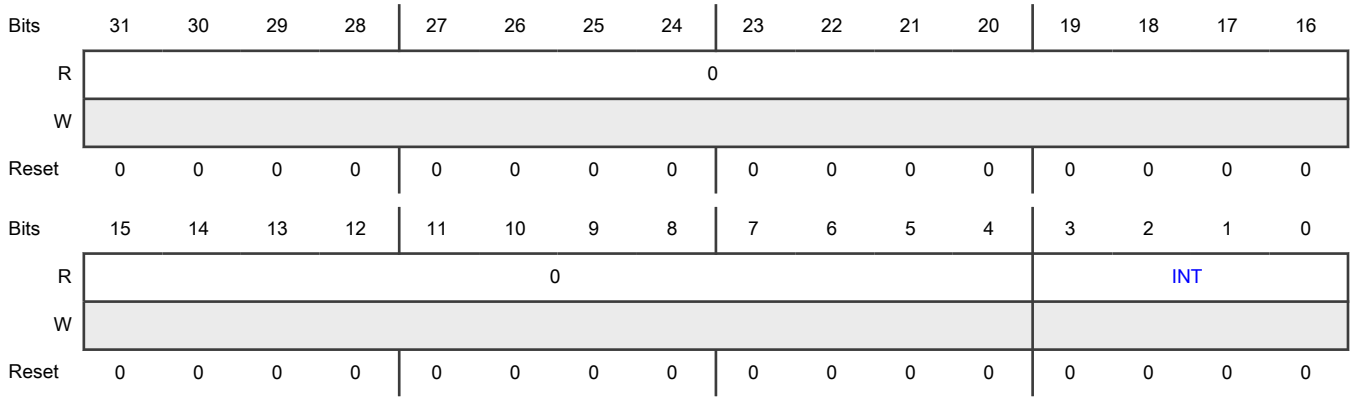
Function

This register shows the current state of the interrupt service requests for all eDMA channels.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram



Fields

Field	Function
31-4 —	Reserved
3-0 INT	Interrupt Request Status The INT register presents the interrupt request status for each eDMA channel. Depending on the appropriate field setting in the transfer control descriptors, the eDMA engine generates an interrupt on data transfer completion or an error condition. The eDMA routes channel interrupt requests to the interrupt controller. During the interrupt service routine associated with any given channel, it is the software's responsibility to clear the appropriate field in the channel's interrupt request register, CHn_INT, thus negating the interrupt request. 0b - Interrupt request for corresponding channel not present 1b - Interrupt request for corresponding channel present

14.6.1.5 Management Page Hardware Request Status (MP_HRS)

Offset

Register	Offset
MP_HRS	Ch

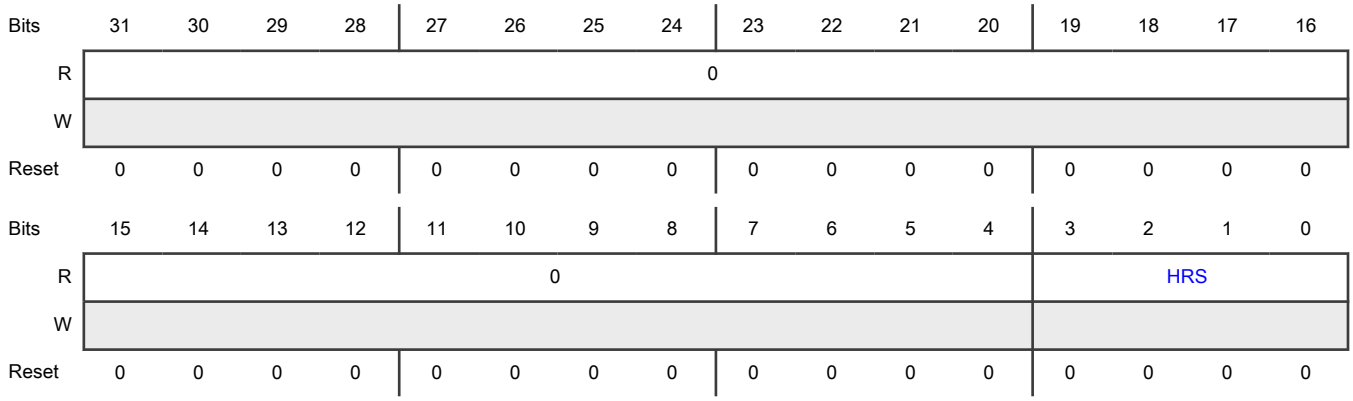
Function

The hardware request status register (HRS) shows the current state of the hardware service request signaling as seen by eDMA's arbitration logic.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram



Fields

Field	Function
31-4 —	Reserved
3-0 HRS	Hardware Request Status The HRS bit for its respective channel remains asserted for the period when a hardware request is present on the channel. 0b - Hardware service request for corresponding channel is not present 1b - Hardware service request for corresponding channel is present

14.6.1.6 Channel Arbitration Group (CH0_GRPRI - CH3_GRPRI)

Offset

Register	Offset
CH0_GRPRI	100h
CH1_GRPRI	104h
CH2_GRPRI	108h
CH3_GRPRI	10Ch

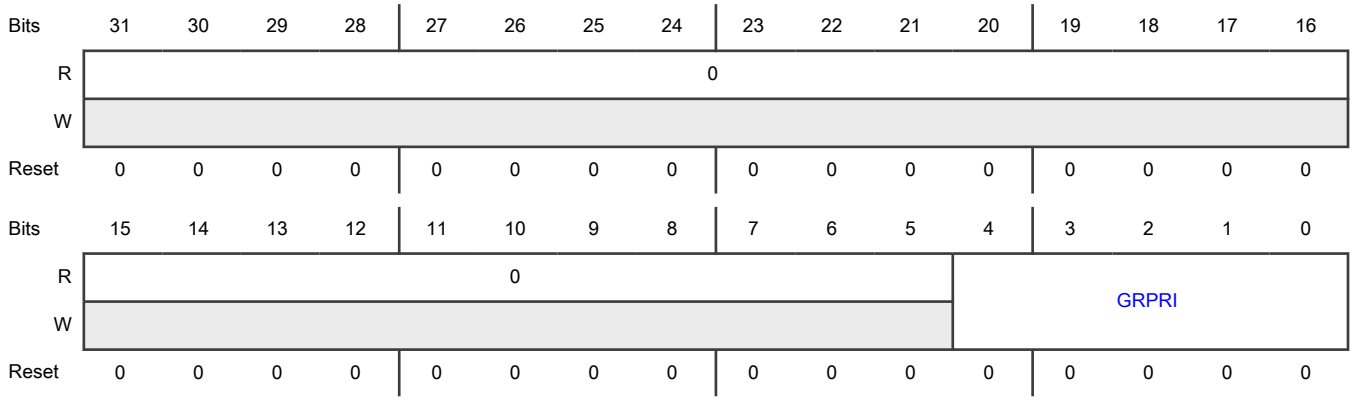
Function

The contents of this register define the arbitration group associated with each channel. Using a fixed-priority group arbitration scheme, eDMA evaluates the arbitration group priorities by numeric value from highest group number to lowest; for example, 0 is the lowest priority, 1 is the next higher priority, then 2, 3, and so on. The range of the group priority values is limited to the values of 0 through 31. Within each arbitration group, the channel priority assignment CH*n*_PRI determines the highest-priority channel.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram



Fields

Field	Function
31-5 —	Reserved
4-0 GRPRI	Arbitration Group For Channel n Fixed-priority arbitration group number.

14.6.2 EDMA_TCD register descriptions

14.6.2.1 EDMA_TCD memory map

eDMA_0_TCD0 base address: 4008_1000h

Offset	Register	Width (In bits)	Access	Reset value
0h - 3000h	Channel Control and Status (CH0_CSR - CH3_CSR)	32	RW	0000_0000h
4h - 3004h	Channel Error Status (CH0_ES - CH3_ES)	32	RW	0000_0000h
8h - 3008h	Channel Interrupt Status (CH0_INT - CH3_INT)	32	RW	0000_0000h
Ch - 300Ch	Channel System Bus (CH0_SBR - CH3_SBR)	32	RW	0000_0005h
10h - 3010h	Channel Priority (CH0_PRI - CH3_PRI)	32	RW	0000_0000h
14h - 3014h	Channel Multiplexor Configuration (CH0_MUX - CH3_MUX)	32	RW	0000_0000h
20h - 3020h	TCD Source Address (TCD0_SADDR - TCD3_SADDR)	32	RW	See section
24h - 3024h	TCD Signed Source Address Offset (TCD0_SOFF - TCD3_SOFF)	16	RW	See section
26h - 3026h	TCD Transfer Attributes (TCD0_ATTR - TCD3_ATTR)	16	RW	See section
28h - 3028h	TCD Transfer Size Without Minor Loop Offsets (TCD0_NBYTES_MLOFFNO - TCD3_NBYTES_MLOFFNO)	32	RW	See section

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Offset	Register	Width (In bits)	Access	Reset value
28h - 3028h	TCD Transfer Size with Minor Loop Offsets (TCD0_NBYTES_MLOFFYES - TCD3_NBYTES_MLOFFYES)	32	RW	See section
2Ch - 302Ch	TCD Last Source Address Adjustment / Store DADDR Address (TCD0_SLAST_SDA - TCD3_SLAST_SDA)	32	RW	See section
30h - 3030h	TCD Destination Address (TCD0_DADDR - TCD3_DADDR)	32	RW	See section
34h - 3034h	TCD Signed Destination Address Offset (TCD0_DOFF - TCD3_DOFF)	16	RW	See section
36h - 3036h	TCD Current Major Loop Count (Minor Loop Channel Linking Disabled) (TCD0_CITER_ELINKNO - TCD3_CITER_ELINKNO)	16	RW	See section
36h - 3036h	TCD Current Major Loop Count (Minor Loop Channel Linking Enabled) (TCD0_CITER_ELINKYES - TCD3_CITER_ELINKYES)	16	RW	See section
38h - 3038h	TCD Last Destination Address Adjustment / Scatter Gather Address (TCD0_DLAST_SGA - TCD3_DLAST_SGA)	32	RW	See section
3Ch - 303Ch	TCD Control and Status (TCD0_CSR - TCD3_CSR)	16	RW	See section
3Eh - 303Eh	TCD Beginning Major Loop Count (Minor Loop Channel Linking Disabled) (TCD0_BITER_ELINKNO - TCD3_BITER_ELINKNO)	16	RW	See section
3Eh - 303Eh	TCD Beginning Major Loop Count (Minor Loop Channel Linking Enabled) (TCD0_BITER_ELINKYES - TCD3_BITER_ELINKYES)	16	RW	See section

14.6.2.2 Channel Control and Status (CH0_CSR - CH3_CSR)

Offset

Register	Offset
CH0_CSR	0h
CH1_CSR	1000h
CH2_CSR	2000h
CH3_CSR	3000h

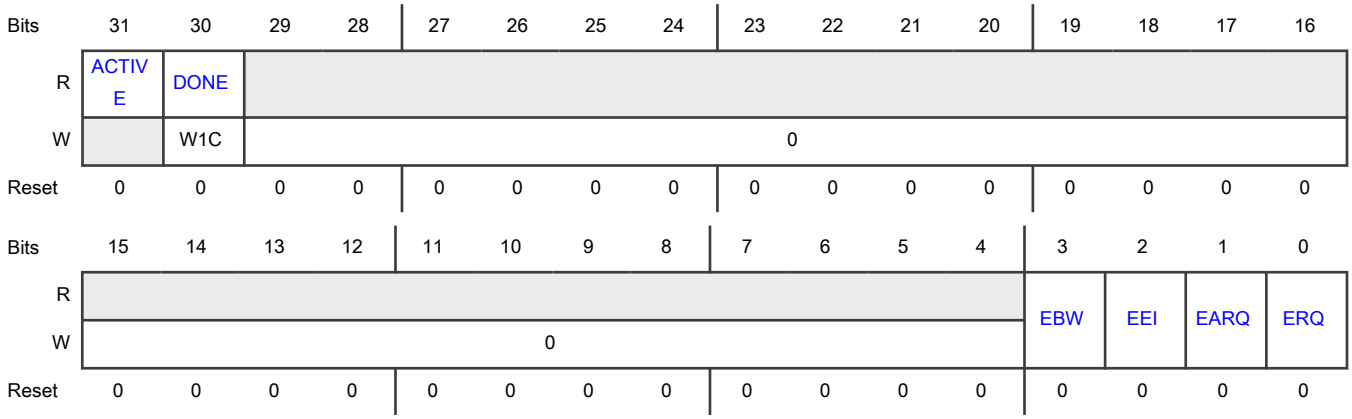
Function

This register contains several fields related to hardware and interrupt requests, configuration, and status for the given channel.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram



Fields

Field	Function
31 ACTIVE	Channel Active The ACTIVE field indicates the channel was selected by arbitration and is executing the prescribed transfers. The eDMA sets it to 1 when channel service begins, and clears it to 0 as the minor loop completes or when any error condition is detected. Except for dynamic scatter/gather or dynamic channel linking, you must not modify the transfer control descriptor when a channel is active.
30 DONE	Channel Done The DONE field indicates the eDMA has completed the major loop. The eDMA engine sets this field as the CITER count reaches zero. If enabled, the eDMA generates an interrupt request corresponding to this completed channel. The software clears it, or the hardware clears it when the channel is activated. **NOTE** • This field must be cleared to 0 before writing the MAJORELINK or ESG fields. • When an error is injected into the eDMA TCD, the eDMA module itself does not detect that the injection has occurred. Based on its configuration, it may generate error interrupts and may or may not assert the DONE signal.
29-4 —	Reserved
3 EBW	Enable Buffered Writes When buffered writes are enabled, all writes except for the last write sequence of the minor loop are signaled by the eDMA as bufferable. 0b - Buffered writes on system bus disabled. Buffered writes on system bus disabled 1b - Buffered writes on system bus enabled. Bufferable write signal asserted on all system bus writes except during last write sequence
2 EEI	Enable Error Interrupt

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	The EEI field enables the error interrupt signal for the channel. The DMA error indicator and the error interrupt enable flag must be asserted before an error interrupt request for a given channel is asserted to the interrupt controller. 0b - Error signal for corresponding channel does not generate error interrupt 1b - Assertion of error signal for corresponding channel generates error interrupt request
1 EARQ	Enable Asynchronous DMA Request The enable asynchronous DMA request field (EARQ) does not affect DMA operations. When set to 1, this field allows the hardware service request enable field (ERQ) to propagate out of the DMA to the power controller. When cleared to 0, this field masks the hardware service request enable field to the power controller. 0b - Disable asynchronous DMA request for the channel 1b - Enable asynchronous DMA request for the channel
0 ERQ	Enable DMA Request Disable a channel's hardware service request at the source before clearing the channel's ERQ field. The DMA hardware request input signal and the enable request field (ERQ) must be asserted before a channel's hardware service request is accepted. The state of the eDMA enable request field does not affect a channel service request made explicitly through software or channel linking. The state of the ERQ field does not affect the channel's START field. 0b - DMA hardware request signal for corresponding channel disabled 1b - DMA hardware request signal for corresponding channel enabled

14.6.2.3 Channel Error Status (CH0_ES - CH3_ES)

Offset

Register	Offset
CH0_ES	4h
CH1_ES	1004h
CH2_ES	2004h
CH3_ES	3004h

Function

The ES provides information concerning the last recorded channel error. Channel errors can be caused by:

- An illegal setting in the transfer control descriptor
- An error termination to a bus initiator read or write cycle

The ERR field signals the presence of an error for the channel. The eDMA engine signals the occurrence of an error condition by setting the appropriate field in this register. The outputs of this register are enabled by the contents of the CHn_CSR[EEI] field, then logically summed across all channels to form an error interrupt request, which may be routed to the interrupt controller. In

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

addition, this enabled error status is logically OR'd onto the channel done interrupt, CHn_INT[INT], thus forming a done or error interrupt on a per channel basis.

During the execution of the interrupt service routine associated with any DMA errors, it is software's responsibility to clear the appropriate bit, negating the error-interrupt request. The normal DMA channel completion indicators (setting the transfer control descriptor DONE flag and the possible assertion of an interrupt request) are not affected when eDMA detects an error. The contents of this ERR register field can also be polled because a non-zero value indicates the presence of a channel error, regardless of the state of the EEI mask.

The state of any given channel's error indicators is affected by writes to this register. Writing a 1 to the ERR field clears the channel's error status, and writing a 0 has no effect.

An unspecified error, where only the ERR field is set to 1, indicates that either a transfer was cancelled with an error. The Management Page Error Status register has full view of the error condition.

See Fault reporting and handling for more details.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	ERR	Reserved														
W	W1C	0														
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved								SAE	SOE	DAE	DOE	NCE	SGE	SBE	DBE
W	0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31 ERR	Error In Channel 0b - An error in this channel has not occurred 1b - An error in this channel has occurred
30-8 —	Reserved
7 SAE	Source Address Error TCDn_SADDR is inconsistent with TCDn_ATTR[SSIZE]. 0b - No source address configuration error 1b - Last recorded error was a configuration error detected in the TCDn_SADDR field
6 SOE	Source Offset Error TCDn_SOFF is inconsistent with TCDn_ATTR[SSIZE].

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	0b - No source offset configuration error 1b - Last recorded error was a configuration error detected in the TCDn_SOFF field
5 DAE	Destination Address Error TCDn_DADDR is inconsistent with TCDn_ATTR[DSIZE]. 0b - No destination address configuration error 1b - Last recorded error was a configuration error detected in the TCDn_DADDR field
4 DOE	Destination Offset Error TCDn_DOFF is inconsistent with TCDn_ATTR[DSIZE]. 0b - No destination offset configuration error 1b - Last recorded error was a configuration error detected in the TCDn_DOFF field
3 NCE	NBYTES/CITER Configuration Error This error indicates that one of the following has occurred: • TCDn_NBYTES is not a multiple of TCDn_ATTR[SSIZE] and TCDn_ATTR[DSIZE] • TCDn_CITER[CITER] is equal to zero • TCDn_CITER[ELINK] is not equal to TCDn_BITER[ELINK] 0b - No NBYTES/CITER configuration error 1b - Last recorded error was a configuration error detected in the TCDn_NBYTES or TCDn_CITER fields
2 SGE	Scatter/Gather Configuration Error When this field is 1, it indicates that TCDn_DLAST_SGA is not on a 32-byte boundary. This field is checked at the beginning of a scatter/gather operation after major loop completion if TCDn_CSR[ESG]is enabled. 0b - No scatter/gather configuration error 1b - Last recorded error was a configuration error detected in the TCDn_DLAST_SGA field
1 SBE	Source Bus Error 0b - No source bus error 1b - Last recorded error was bus error on source read
0 DBE	Destination Bus Error 0b - No destination bus error 1b - Last recorded error was bus error on destination write

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

14.6.2.4 Channel Interrupt Status (CH0_INT - CH3_INT)

Offset

Register	Offset
CH0_INT	8h
CH1_INT	1008h
CH2_INT	2008h
CH3_INT	3008h

Function

The INT field signals the presence of an interrupt request for the channel. Depending on the appropriate bit setting in the transfer control descriptors, the eDMA engine generates an interrupt on data transfer completion or an error condition.

The outputs of this register are directly routed to the interrupt controller. During the interrupt service routine associated with any given channel, it is the software's responsibility to clear the appropriate bit, negating the interrupt request. On writes to INT, a 1 clears the channel's interrupt request. A zero has no effect on the channel's current interrupt status.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															INT
W																W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-1 —	Reserved
0 INT	Interrupt Request 0b - Interrupt request for corresponding channel cleared 1b - Interrupt request for corresponding channel active

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

14.6.2.5 Channel System Bus (CH0_SBR - CH3_SBR)

Offset

Register	Offset
CH0_SBR	Ch
CH1_SBR	100Ch
CH2_SBR	200Ch
CH3_SBR	300Ch

Function

The Channel System Bus register places identification and attribute information on the system bus interface for the eDMA.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															EMI
W	0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PAL	0											MID			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

Fields

Field	Function
31-17 —	Reserved
16 EMI	Enable Initiator ID Replication The eDMA initiator ID replication field allows the eDMA to use the same protection level and system bus ID of the initiator programming the eDMA's TCD. When enabled, the eDMA uses the initiator ID and protection level stored in the CHn_SBR registers, instead of the eDMA's default values. When a initiator (for example a core) programs a TCD, its initiator ID is captured when the TCD*n*_CSR control attributes are written. A scatter/gather operation does not affect the CHn_SBR registers. You can write the EMI only if MP_CSR[GMRC] = 1, which means Global Initiator ID Replication Control is enabled; otherwise, the EMI is forced to zero. **NOTE** If initiator ID replication is disabled, the nonsecure, user protection level for DMA transfers is used.

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	0b - Initiator ID replication is disabled 1b - Initiator ID replication is enabled
15 PAL	Privileged Access Level This field controls DMA's protection level on the system bus when the channel is active. **NOTE** When you enable initiator ID replication, the value captured in this register is the privilege level of the core or other initiator writing the channel's transfer control descriptor, which is the lower byte of TCD*n*_CSR. 0b - User protection level for DMA transfers 1b - Privileged protection level for DMA transfers
14-4 —	Reserved
3-0 MID	Initiator ID This field controls the DMA's initiator ID on the system bus when the channel is active. **NOTE** The ID captured in this register reflects the initiator ID of the core or other initiator writing the channel's control attributes, which are in the lower byte of TCD*n*_CSR.

14.6.2.6 Channel Priority (CH0_PRI - CH3_PRI)

Offset

Register	Offset
CH0_PRI	10h
CH1_PRI	1010h
CH2_PRI	2010h
CH3_PRI	3010h

Function

The contents of these registers define unique priorities associated with each channel within the same channel group. Channel grouping is programmed via Channel Arbitration Group (CH0_GRPRI - CH3_GRPRI).

The channel priorities within a group are evaluated by numeric value; for example, 0 is the lowest priority, 1 is the next higher priority, then 2, 3, and so on. Software must program the channel priorities with unique values; otherwise, channel numbers with the same, non-zero value, will be selected based on channel number with the higher channel number having higher priority.

If more than one channel in a group has an arbitration priority level value of zero, then the arbitration mode field MP_CSR[ERCA] is used to determine the arbitration scheme for all channels with APL=0 within a group.

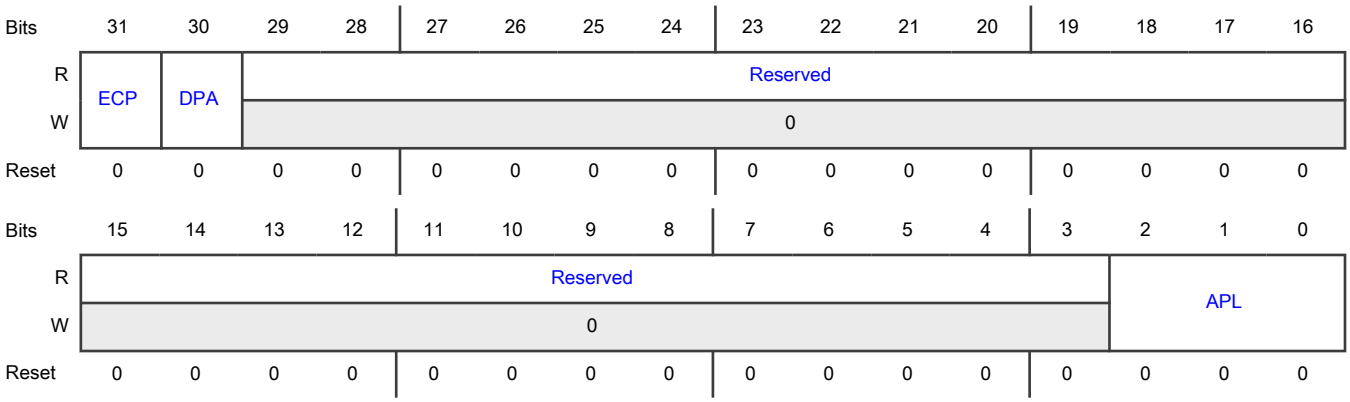
All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

When you enable round-robin channel arbitration (MP_CSR[ERCA] = 1), all channels with APL=0 within a group will use a round-robin arbitration scheme, which rotates among these channels requesting service without regard to priority. Round-robin provides a fairness mechanism within an arbitration group.

When you enable fixed-priority channel arbitration (MP_CSR[ERCA] = 0), eDMA selects channels with APL=0 based on channel number, with the higher channel number having higher priority.

Diagram



Fields

Field	Function
31 ECP	Enable Channel Preemption 0b - Channel cannot be suspended by a higher-priority channel's service request 1b - Channel can be temporarily suspended by a higher-priority channel's service request
30 DPA	Disable Preempt Ability 0b - Channel can suspend a lower-priority channel 1b - Channel cannot suspend any other channel, regardless of channel priority
29-3 —	Reserved
2-0 APL	Arbitration Priority Level Channel priority level for arbitration within the assigned arbitration group.

14.6.2.7 Channel Multiplexor Configuration (CH0_MUX - CH3_MUX)

Offset

Register	Offset
CH0_MUX	14h
CH1_MUX	1014h

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Register	Offset
CH2_MUX	2014h
CH3_MUX	3014h

Function

Each of the DMA channels can be independently associated with various peripherals in the system. The Channel Multiplexor Configuration register selects the peripheral assigned to each channel. Service requests from the peripheral should be disabled when configuring a channel to a peripheral source.

Each channel must have a unique value when selecting a peripheral slot in the channel mux configuration. The only value that may overlap is source 0. If there is an attempt to write a mux configuration value that is already consumed by any channel, a mux configuration of 0 (SRC = 0) will be written.

All channels will default to source 0. When a particular peripheral is needed, the channel's mux configuration is set to that source number. When the peripheral is no longer needed, the mux configuration for that channel should be written to 0, thus releasing the resource.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W	0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved								SRC							
W	0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-7 —	Reserved
6-0 SRC	Service Request Source Hardware service request source for the channel. **NOTE** With the exception of 0, attempts to write a value already in use will be forced to 0.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

14.6.2.8 TCD Source Address (TCD0_SADDR - TCD3_SADDR)

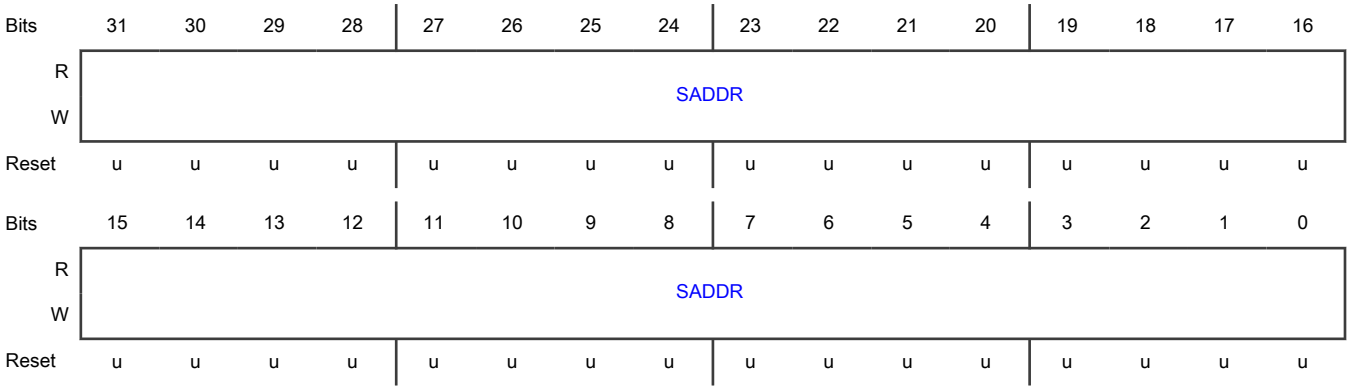
Offset

Register	Offset
TCD0_SADDR	20h
TCD1_SADDR	1020h
TCD2_SADDR	2020h
TCD3_SADDR	3020h

Function

This register contains the address for the read transactions.

Diagram



Fields

Field	Function
31-0 SADDR	Source Address Memory address pointing to the source data.

14.6.2.9 TCD Signed Source Address Offset (TCD0_SOFF - TCD3_SOFF)

Offset

Register	Offset
TCD0_SOFF	24h
TCD1_SOFF	1024h
TCD2_SOFF	2024h
TCD3_SOFF	3024h

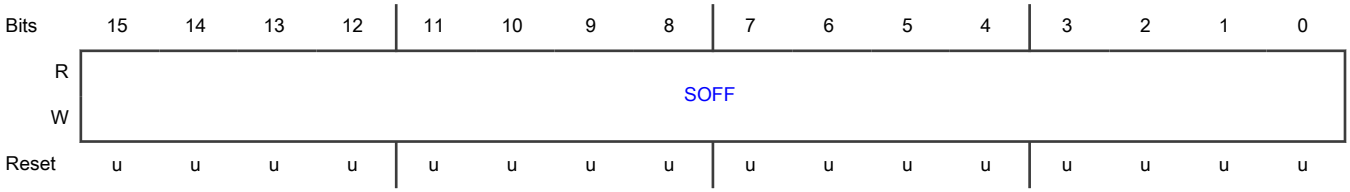
All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Function

This register contains the sign-extended value added to Source Address register after each read transaction.

Diagram



Fields

Field	Function
15-0 SOFF	Source Address Signed Offset Sign-extended offset applied to the current source address to form the next-state value as each source read is completed.

14.6.2.10 TCD Transfer Attributes (TCD0_ATTR - TCD3_ATTR)

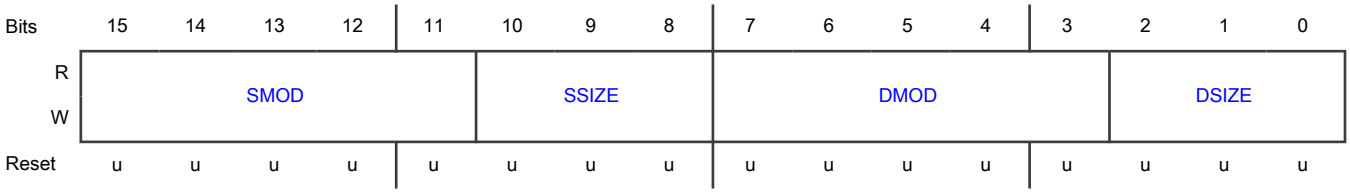
Offset

Register	Offset
TCD0_ATTR	26h
TCD1_ATTR	1026h
TCD2_ATTR	2026h
TCD3_ATTR	3026h

Function

This register contains size and option modulo addressing information for source and destination addresses.

Diagram



Fields

Field	Function
15-11	Source Address Modulo

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
SMOD	This field defines a specific address range, which is the value after the SADDR + SOFF calculation is performed on the original register value. Setting this field makes it easy to implement a circular data queue. For data queues requiring power-of-2-sized bytes, the queue must start at a 0-modulo-size address and the SMOD field must be set to the appropriate value for the queue, freezing the required number of upper address bits. The value programmed into this field specifies the number of lower address bits that are allowed to change. For a circular queue application, you typically set TCDn_SOFF[SOFF] to the transfer size to implement post-increment addressing, with the SMOD function constraining the addresses to a 0-modulo-size range. 0_0000b - Source address modulo feature disabled
10-8 SSIZE	Source Data Transfer Size 000b - 8-bit 001b - 16-bit 010b - 32-bit 011b - 64-bit 100b - 16-byte 101b - 32-byte 110b - Reserved 111b - Reserved
7-3 DMOD	Destination Address Modulo See the SMOD definition.
2-0 DSIZE	Destination Data Transfer Size See the SSIZE definition.

14.6.2.11 TCD Transfer Size Without Minor Loop Offsets (TCD0_NBYTES_MLOFFNO - TCD3_NBYTES_MLOFFNO)

Offset

Register	Offset
TCD0_NBYTES_MLOFFNO	28h
TCD1_NBYTES_MLOFFNO	1028h
TCD2_NBYTES_MLOFFNO	2028h
TCD3_NBYTES_MLOFFNO	3028h

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Function

The TCDn_NBYTES field defines the number of bytes to transfer per service request.

Minor loop offsets are address offset values added to the final source address (TCDn_SADDR), or destination address (TCDn_DADDR), upon minor loop completion. Minor loop completion is when the channel has finished the service request and has transferred NBYTES. When minor loop offsets are enabled, the minor loop offset value (TCDn_NBYTES_MLOFFYES[MLOFF]) is added to the final source address (TCDn_SADDR), to the final destination address (TCDn_DADDR), or to both, prior to the addresses being written back to the TCD. If the major loop is complete, the minor loop offset is ignored and the major loop address offsets (TCDn_SLAST_SDA and TCDn_DLAST_SGA) are used to compute the next TCDn_SADDR and TCDn_DADDR values.

When minor loop mapping is enabled (SMLOE or DMLOE is 1), TCDn_NBYTES_MLOFFNO/TCDn_NBYTES_MLOFFYES is redefined. A portion of TCDn_NBYTES_MLOFFNO/TCDn_NBYTES_MLOFFYES is used to specify multiple fields:

- A source enable bit (SMLOE) to specify the minor loop offset must be applied to the source address (TCDn_SADDR) upon minor loop completion
- A destination enable bit (DMLOE) to specify the minor loop offset must be applied to the destination address (TCDn_DADDR) upon minor loop completion
- The sign extended minor loop offset value (MLOFF)

The same offset value (MLOFF) is used for both source and destination minor loop offsets. When either minor loop offset is enabled (SMLOE set or DMLOE set), the NBYTES field is reduced to 10 bits. If both minor loop offsets are disabled (SMLOE cleared and DMLOE cleared), the NBYTES field is a 30-bit vector.

One of two register profiles (this register or TCDn_NBYTES_MLOFFYES), defines the number of bytes to transfer per request. Which register to use depends on whether source or destination minor loop mapping is enabled.

TCDn_NBYTES_MLOFFNO/TCDn_NBYTES_MLOFFYES is defined as follows:

- If SMLOE = 0 and DMLOE = 0, then see the TCDn_NBYTES_MLOFFNO register description.
- If either SMLOE or DMLOE is 1, then see the TCDn_NBYTES_MLOFFYES register description.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	SMLOE	DMLOE	NBYTES													
W																
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	NBYTES															
W																
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

Fields

Field	Function
31 SMLOE	Source Minor Loop Offset Enable Selects whether the minor loop offset is applied to the source address upon minor loop completion.

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	0b - Minor loop offset not applied to SADDR 1b - Minor loop offset applied to SADDR
30 DMLOE	Destination Minor Loop Offset Enable Selects whether the minor loop offset is applied to the destination address upon minor loop completion. 0b - Minor loop offset not applied to DADDR 1b - Minor loop offset applied to DADDR
29-0 NBYTES	Number of Bytes To Transfer Per Service Request Number of bytes to be transferred for each service request of the channel. When a channel activates, the module loads the appropriate TCD contents into the eDMA engine and performs the appropriate reads and writes until the byte transfer count has been reached. This process is normally an indivisible operation and cannot be halted. It can, however, be stalled by using the bandwidth control field, or via preemption. After the byte count is exhausted, the SADDR and DADDR values are written back into the TCD memory, and the major loop iteration count (CITER) is decremented by one and written back to the TCD memory. If the major iteration count is complete, additional processing is performed.

14.6.2.12 TCD Transfer Size with Minor Loop Offsets (TCD0_NBYTES_MLOFFYES - TCD3_NBYTES_MLOFFYES)

Offset

Register	Offset
TCD0_NBYTES_MLOFFYES	28h
TCD1_NBYTES_MLOFFYES	1028h
TCD2_NBYTES_MLOFFYES	2028h
TCD3_NBYTES_MLOFFYES	3028h

Function

The TCDn_NBYTES field defines the number of bytes to transfer per service request.

Minor loop offset is an address offset value added to the final source address (TCDn_SADDR) or destination address (TCDn_DADDR) upon minor loop completion. Minor loop completion occurs when the channel has finished the service request and has transferred NBYTES. Minor loop offsets are enabled by setting either the source enable bit (SMLOE) or the destination enable bit (DMLOE).

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

The source enable bit (SMLOE) specifies the minor loop offset value (MLOFF) that is to be applied to the source address (TCDn_SADDR) upon minor loop completion. The destination enable bit (DMLOE) specifies the minor loop offset (MLOFF) that is to be applied to the destination address (TCDn_DADDR) upon minor loop completion.

If the major loop is complete, the minor loop offsets are ignored and the major loop address offsets (TCDn_SLAST_SDA and TCDn_DLAST_SGA) are used to compute the next TCDn_SADDR and TCDn_DADDR values.

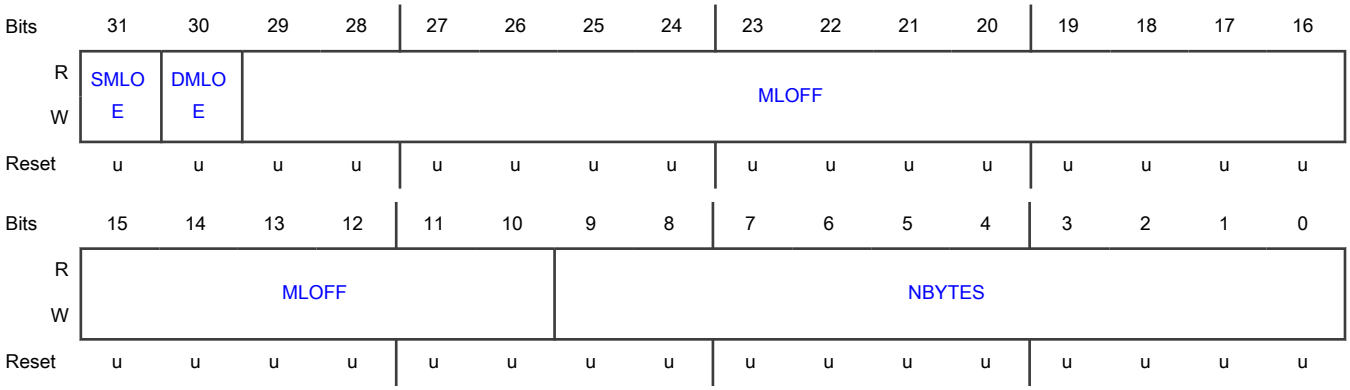
When you enable the minor loop offset overlay (either SMLOE or DMLOE is 1), eDMA redefines TCDn_NBYTES_MLOFFNO/TCDn_NBYTES_MLOFFYES. A portion of TCDn_NBYTES_MLOFFNO/TCDn_NBYTES_MLOFFYES specifies the sign-extended minor loop offset value (MLOFF). The same offset value (MLOFF) applies to both source and destination minor loop offsets. When the minor loop offset is enabled, you must align it to the transfer size of the source or destination it is associated with. When either minor loop offset is enabled (SMLOE set or DMLOE set), the NBYTES field is reduced to 10 bits. If both minor loop offsets are disabled (SMLOE cleared and DMLOE cleared), the NBYTES field is a 30-bit vector.

One of two register profiles (this register or TCDn_NBYTES_MLOFFNO) defines the number of bytes to transfer per request. Which register to use depends on whether source or destination minor loop mapping is enabled.

TCDn_NBYTES_MLOFFYES is defined as follows:

- If either minor loop offset is enabled (SMLOE or DMLOE = 1), then see the TCDn_NBYTES_MLOFFYES register description.
- If SMLOE and DMLOE are both 0, then see the TCDn_NBYTES_MLOFFNO register description.

Diagram



Fields

Field	Function
31 SMLOE	Source Minor Loop Offset Enable Selects whether the minor loop offset is applied to the source address upon minor loop completion. 0b - Minor loop offset not applied to SADDR 1b - Minor loop offset applied to SADDR
30 DMLOE	Destination Minor Loop Offset Enable Selects whether the minor loop offset is applied to the destination address upon minor loop completion. 0b - Minor loop offset not applied to DADDR 1b - Minor loop offset applied to DADDR
29-10	Minor Loop Offset

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
MLOFF	If SMLOE or DMLOE is 1, this field represents a sign-extended offset applied to the source or destination address to form the next-state value after the minor loop completes.
9-0 NBYTES	Number of Bytes To Transfer Per Service Request The number of bytes to be transferred in each service request of the channel. As a channel activates, the module loads the appropriate TCD contents into the eDMA engine and performs the appropriate reads and writes until the minor byte transfer count has been reached. This is an indivisible operation and cannot be halted. It can, however, be stalled by using the bandwidth control field, or via preemption. After the minor count is exhausted, the SADDR and DADDR values are written back into the TCD memory, and the major iteration count is decremented and restored to the TCD memory. If the major iteration count is complete, additional processing is performed.

14.6.2.13 TCD Last Source Address Adjustment / Store DADDR Address (TCD0_SLAST_SDA - TCD3_SLAST_SDA)

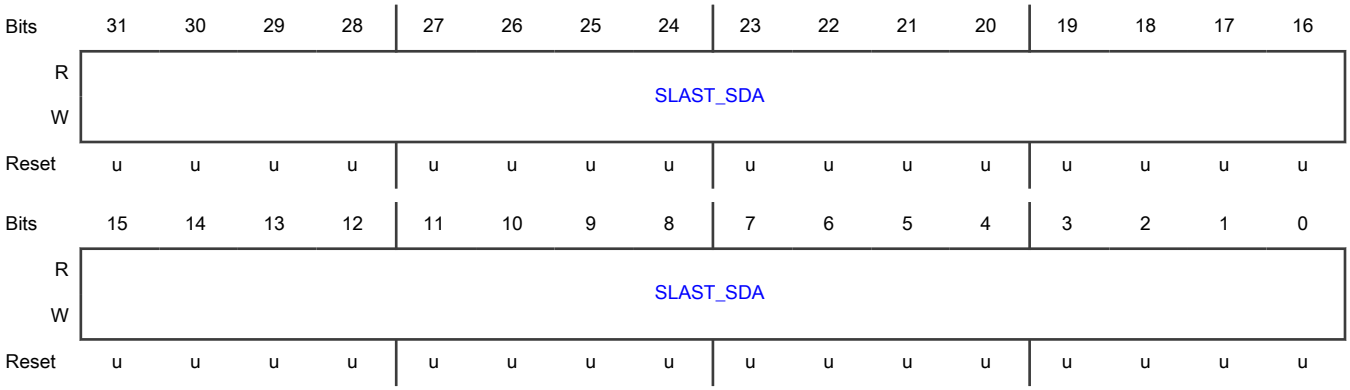
Offset

Register	Offset
TCD0_SLAST_SDA	2Ch
TCD1_SLAST_SDA	102Ch
TCD2_SLAST_SDA	202Ch
TCD3_SLAST_SDA	302Ch

Function

This register contains the value added to the source address when the major loop is complete. When the store destination address option is enabled, this field provides a pointer to memory for storing the final destination address.

Diagram



All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Fields

Field	Function
31-0 SLAST_SDA	Last Source Address Adjustment / Store DADDR Address Source last address adjustment or the system memory address for destination address (DADDR) storage. If (TCDn_CSR[ESDA] = 0), then: • Adjustment value is added to the source address at the completion of the major iteration count. This value can be used to restore the source address to the initial value or adjust the address to reference the next data structure. • This field uses two's complement notation for the final source address adjustment. Otherwise: • This address points to the 32-bit-aligned memory location where the destination address (DADDR) is to be stored in system memory. By saving the final destination address in system memory via the ESDA feature, you are able to compute the size of a variable destination data buffer by simply subtracting the beginning DADDR from the final, saved DADDR. This feature is used together with the scatter/gather operation to prevent the loss of the final DADDR, which is overwritten during the scatter/gather operation. The "Store Destination Address" (SDA) value must be a 32-bit-aligned location because the eDMA forces the lower two address bits of the SLAST_SDA field to zero when ESDA is enabled. The module performs this write operation when the major loop is done; that is, when the major iteration count (CITER) decrements to zero.

14.6.2.14 TCD Destination Address (TCD0_DADDR - TCD3_DADDR)

Offset

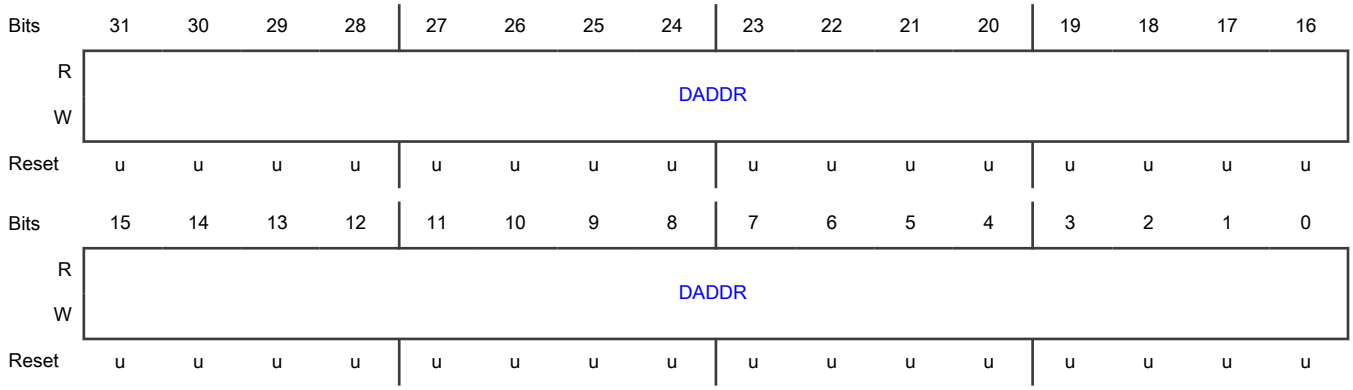
Register	Offset
TCD0_DADDR	30h
TCD1_DADDR	1030h
TCD2_DADDR	2030h
TCD3_DADDR	3030h

Function
This register contains the address for the write transactions.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram



Fields

Field	Function
31-0 DADDR	Destination Address Memory address pointing to the destination data.

14.6.2.15 TCD Signed Destination Address Offset (TCD0_DOFF - TCD3_DOFF)

Offset

Register	Offset
TCD0_DOFF	34h
TCD1_DOFF	1034h
TCD2_DOFF	2034h
TCD3_DOFF	3034h

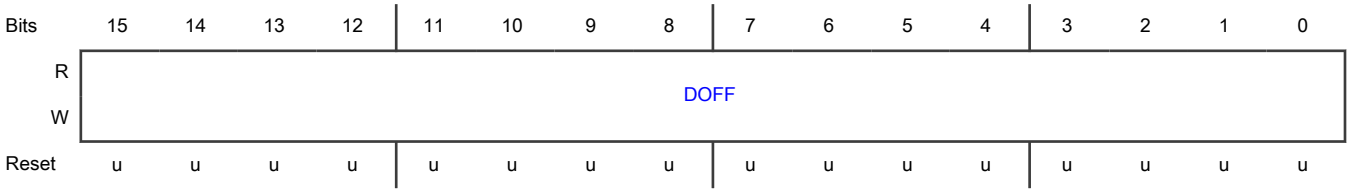
Function

This register contains the sign-extended value added to Destination Address register after each write transaction.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram



Fields

Field	Function
15-0 DOFF	Destination Address Signed Offset Sign-extended offset that is applied to the current destination address to form the next-state value as each destination write is completed.

14.6.2.16 TCD Current Major Loop Count (Minor Loop Channel Linking Disabled) (TCD0_CITER_ELINKNO - TCD3_CITER_ELINKNO)

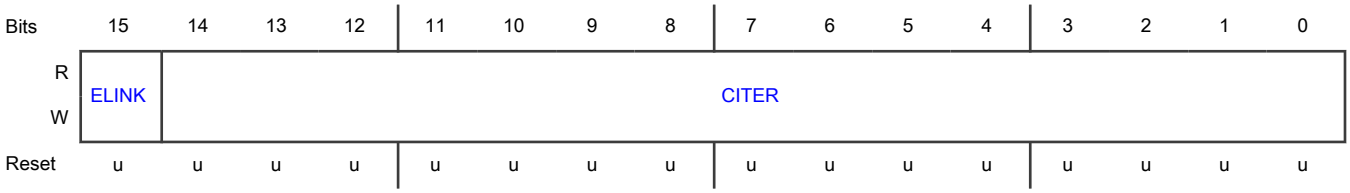
Offset

Register	Offset
TCD0_CITER_ELINKNO	36h
TCD1_CITER_ELINKNO	1036h
TCD2_CITER_ELINKNO	2036h
TCD3_CITER_ELINKNO	3036h

Function

If TCDn_CITER[ELINK] is 0, the TCDn_CITER register is defined as follows.

Diagram



Fields

Field	Function
15 ELINK	Enable Link

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	As the channel completes the minor loop, this flag enables linking to another channel as defined by the relevant LINKCH field. The link target channel initiates a channel service request via an internal mechanism that sets the TCDn_CSR[START] bit of the specified channel to 1. If channel linking is disabled, the CITER value is extended to 15 bits in place of a link channel number. If the major loop is exhausted, this link mechanism is suppressed in favor of MAJORELINK channel linking. **NOTE** This field must be equal to the BITER[ELINK] field; otherwise, a configuration error is reported. 0b - Channel-to-channel linking disabled 1b - Channel-to-channel linking enabled
14-0 CITER	Current Major Iteration Count This 9-bit (ELINK = 1) or 15-bit (ELINK = 0) count represents the current major loop count for the channel. It is decremented each time the channel finishes a service request and is written back to TCD memory. After the major iteration count is exhausted, the channel performs a number of operations — for example, final source and destination address calculations — and optionally generates an interrupt to signal channel completion before reloading the CITER field from the Beginning Iteration Count (BITER) field. **NOTE** When the CITER field is initially loaded by software, it must be set to the same value as that contained in the BITER field. **NOTE** If the channel is configured to execute a single service request, the initial values of BITER and CITER should be 0x0001.

14.6.2.17 TCD Current Major Loop Count (Minor Loop Channel Linking Enabled) (TCD0_CITER_ELINKYES - TCD3_CITER_ELINKYES)

Offset

Register	Offset
TCD0_CITER_ELINKYES	36h
TCD1_CITER_ELINKYES	1036h
TCD2_CITER_ELINKYES	2036h
TCD3_CITER_ELINKYES	3036h

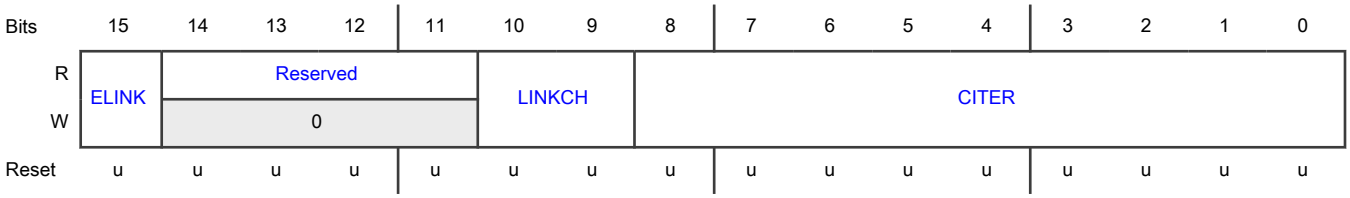
All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Function

If TCDn_CITER[ELINK] is 1, the TCDn_CITER register is defined as follows.

Diagram



Fields

Field	Function
15 ELINK	Enable Link As the channel completes the minor loop, this flag enables linking to another channel as defined by the relevant LINKCH field. When enabled, an internal mechanism sets the TCDn_CSR[START] field of the specified channel (LINKCH) upon minor loop completion. If channel linking is disabled, the CITER value is extended to 15 bits in place of a link channel number. If the major loop is exhausted, this link mechanism is suppressed in favor of MAJORELINK channel linking. **NOTE** This field must be equal to the BITER[ELINK] field; otherwise, a configuration error is reported. 0b - Channel-to-channel linking disabled 1b - Channel-to-channel linking enabled
14-11 —	Reserved
10-9 LINKCH	Minor Loop Link Channel Number If channel-to-channel linking is enabled (ELINK = 1), then after the minor loop is exhausted the eDMA engine initiates a channel service request to the channel defined by this field by writing that channel's TCDn_CSR[START] field to 1.
8-0 CITER	Current Major Iteration Count This 9-bit (ELINK = 1) or 15-bit (ELINK = 0) count represents the current major loop count for the channel. It is decremented each time the channel finishes a service request and is written back to the TCD memory. After the major iteration count is exhausted, the channel performs a number of operations — for example, final source and destination address calculations — and optionally generates an interrupt to signal channel completion before reloading the CITER field from the Beginning Iteration Count (BITER) field. **NOTE** When the CITER field is initially loaded by software, it must be set to the same value as that contained in the BITER field.

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	NOTE If the channel is configured to execute a single service request, the initial values of BITER and CITER should be 0x0001.

14.6.2.18 TCD Last Destination Address Adjustment / Scatter Gather Address (TCD0_DLAST_SGA - TCD3_DLAST_SGA)

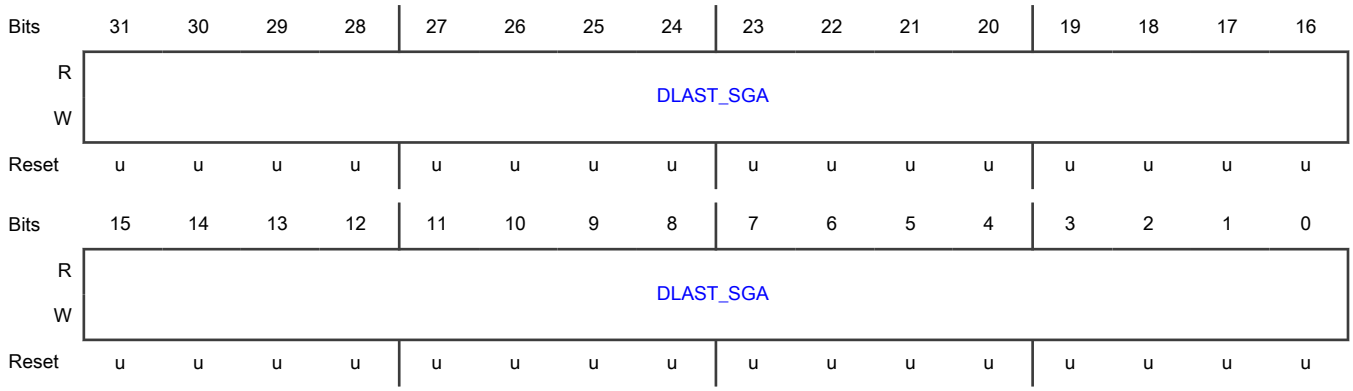
Offset

Register	Offset
TCD0_DLAST_SGA	38h
TCD1_DLAST_SGA	1038h
TCD2_DLAST_SGA	2038h
TCD3_DLAST_SGA	3038h

Function

This register contains the value added to the destination address when the major loop is complete. When the Scatter/Gather option is enabled, this field provides a pointer to memory for fetching a transfer control descriptor to reprogram the channel.

Diagram



Fields

Field	Function
31-0 DLAST_SGA	Last Destination Address Adjustment / Scatter Gather Address Adjustment of the last destination address or the memory address for the next transfer control descriptor to be loaded into this channel (scatter/gather). If (TCDn_CSR[ESG] = 0) then:

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Field	Function
	• Adjustment value is added to the destination address at the completion of the major iteration count. This value can apply to restore the destination address to the initial value or adjust the address to reference the next data structure. • This field uses two's complement notation for the final destination address adjustment. Otherwise: • This address points to the beginning of a 0-modulo 32-byte region containing the next transfer control descriptor to be loaded into this channel. This channel reload is performed as the major iteration count completes. The scatter/gather address must be 0-modulo 32-byte, or else a configuration error is reported.

14.6.2.19 TCD Control and Status (TCD0_CSR - TCD3_CSR)

Offset

Register	Offset
TCD0_CSR	3Ch
TCD1_CSR	103Ch
TCD2_CSR	203Ch
TCD3_CSR	303Ch

Function
This register is used to enable optional features.

Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	BWC				MAJORLINKCH				ESDA	EEOP	MAJOREL...	ESG	DREQ	INTHALF	INTMAJOR	START
W			0													
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	0

Fields

Field	Function
15-14 BWC	Bandwidth Control Throttles the amount of bus bandwidth consumed by the eDMA. Generally, as the eDMA processes the minor loop, it continuously generates read/write sequences until the minor count is exhausted. This field forces eDMA to stall after the completion of each read/write access, to control the bus request bandwidth seen by the system bus interconnect.

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	NOTE If the source and destination sizes are equal, this field is ignored between the first and second transfers and after the last write of each minor loop. This behavior is a side effect of reducing start-up latency. 00b - No eDMA engine stalls 01b - Reserved 10b - eDMA engine stalls for 4 cycles after each R/W 11b - eDMA engine stalls for 8 cycles after each R/W
13-10 —	Reserved
9-8 MAJORLINKCH	Major Loop Link Channel Number If (MAJORELINK = 0) then: • No channel-to-channel linking, or chaining, is performed after the major loop counter is exhausted. Otherwise: • After the major loop counter is exhausted, the eDMA engine initiates a channel service request at the channel defined by this field by setting that channel's TCDn_CSR[START] field to 1.
7 ESDA	Enable Store Destination Address As the channel completes the major loop by either the current iteration counter (CITER) decrementing to 0, or by receiving an enabled end-of-packet signal, this field enables writing the destination address (DADDR) to the address stored in the SLAST_SDA field. The value written to system memory is the next DADDR value prior to the DLAST_SGA offset being applied, or overwritten by an enabled scatter/gather operation. When the ESDA bit is 1, SLAST_SDA contains the write pointer instead of the final source address offset. Because this is a pointer and not a final offset, a last source address offset of zero is applied to SADDR instead of the SLAST_SGA value. 0b - Ability to store destination address to system memory disabled 1b - Ability to store destination address to system memory enabled
6 EEOP	Enable End-Of-Packet Processing When enabled by the EEOP field, an end-of-packet hardware input signal directs eDMA to discontinue executing the active channel, and to treat the shutdown as the major-loop-completed event. If the EEOP field is 1, the end-of-packet signal from supported peripherals is accepted. If the EEOP field is 0, the end-of-packet input is ignored. With an end-of-packet retirement, the current TCD destination address (or ESDA-saved destination address), minus the software-saved initial address (DADDR), reflects the total amount of data transferred. 0b - End-of-packet operation disabled 1b - End-of-packet hardware input signal enabled
5	Enable Link When Major Loop Complete

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
MAJORELINK	As the channel completes the major loop, this flag enables linking to another channel defined by MAJORLINKCH. The link target channel initiates a channel service request via an internal mechanism that sets the TCDn_CSR[START] field of the specified channel. **NOTE** To support the dynamic linking coherency model, this field is forced to 0 if written when TCDn_CSR[DONE] is 1. 0b - Channel-to-channel linking disabled 1b - Channel-to-channel linking enabled
4 ESG	Enable Scatter/Gather Processing As the channel completes the major loop, this flag enables scatter/gather processing in the current channel. If enabled, the eDMA engine uses TCDn_DLAST_SGA as a memory pointer to a 0-modulo 32-bit address containing a 32-byte data structure, which is loaded as the transfer control descriptor into local memory. **NOTE** To support the dynamic scatter/gather coherency model, this field is forced to 0 if written when TCDn_CSR[DONE] is 1. 0b - Current channel's TCD is normal format 1b - Current channel's TCD specifies scatter/gather format.
3 DREQ	Disable Request If this flag is 1, the eDMA hardware automatically clears the corresponding ERQ bit when the current major iteration count reaches 0. 0b - No operation. Channel's ERQ field not affected 1b - Clear the ERQ field to 0 upon major loop completion, thus disabling hardware service requests. Channel's ERQ field cleared to 0 when major loop complete
2 INTHALF	Enable Interrupt If Major Counter Half-complete If this flag is 1, the channel generates an interrupt request by setting the appropriate field in the INT register to 1 when the current major iteration count reaches the halfway point. Specifically, the comparison performed by the eDMA engine is (CITER = (BITER/2)). This halfway point interrupt request is provided to support double-buffered, also known as ping-pong, schemes, or other types of data movement where the processor needs an early indication of the transfer's progress. **NOTE** If BITER = 1, do not use INTHALF; use INTMAJOR instead. 0b - Halfway point interrupt disabled 1b - Halfway point interrupt enabled
1 INTMAJOR	Enable Interrupt If Major count complete If this flag is 1, the channel generates an interrupt request by setting the appropriate field in the INT register to 1 when the current major iteration count (CITER) reaches 0.

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	0b - End-of-major loop interrupt disabled 1b - End-of-major loop interrupt enabled
0 START	Channel Start If this flag is 1, the channel is requesting service. The eDMA hardware automatically clears this flag to 0 after the channel begins execution. 0b - Channel not explicitly started 1b - Channel explicitly started via a software-initiated service request

14.6.2.20 TCD Beginning Major Loop Count (Minor Loop Channel Linking Disabled) (TCD0_BITER_ELINKNO - TCD3_BITER_ELINKNO)

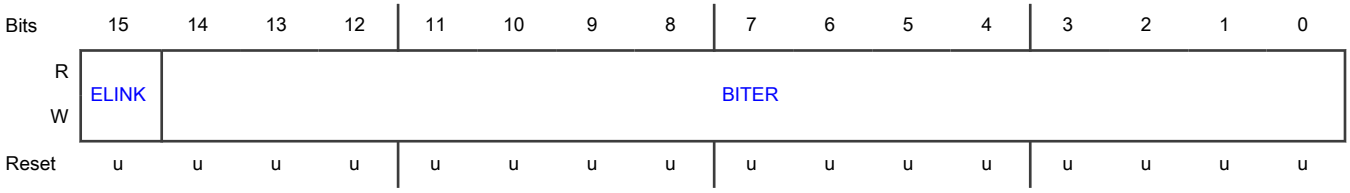
Offset

Register	Offset
TCD0_BITER_ELINKNO	3Eh
TCD1_BITER_ELINKNO	103Eh
TCD2_BITER_ELINKNO	203Eh
TCD3_BITER_ELINKNO	303Eh

Function

If the TCDn_BITER[ELINK] field is 0, the TCDn_BITER register is defined as follows.

Diagram



Fields

Field	Function
15 ELINK	Enables Link As the channel completes the minor loop, this flag enables linking to another channel as defined by BITER[LINKCH]. The link target channel initiates a channel service request via an internal mechanism that sets the TCDn_CSR[START] field of the specified channel. If channel linking is disabled, the BITER value extends to 15 bits in place of a link channel number. If the major loop is exhausted, this link mechanism is suppressed in favor of the MAJORELINK channel linking.

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	NOTE When the software loads the TCD, this field must be set equal to the corresponding CITER field; otherwise, a configuration error is reported. As the major iteration count is exhausted, eDMA reloads the contents of this field into the CITER field. 0b - Channel-to-channel linking disabled 1b - Channel-to-channel linking enabled
14-0 BITER	Starting Major Iteration Count As the transfer control descriptor is first loaded by software, this 9-bit (ELINK = 1) or 15-bit (ELINK = 0) field must be set equal to the value in the CITER field. As the major iteration count is exhausted, eDMA reloads the contents of this field into the CITER field. If the channel is configured to execute a single service request, the initial values of BITER and CITER must be 0x0001.

14.6.2.21 TCD Beginning Major Loop Count (Minor Loop Channel Linking Enabled) (TCD0_BITER_ELINKYES - TCD3_BITER_ELINKYES)

Offset

Register	Offset
TCD0_BITER_ELINKYES	3Eh
TCD1_BITER_ELINKYES	103Eh
TCD2_BITER_ELINKYES	203Eh
TCD3_BITER_ELINKYES	303Eh

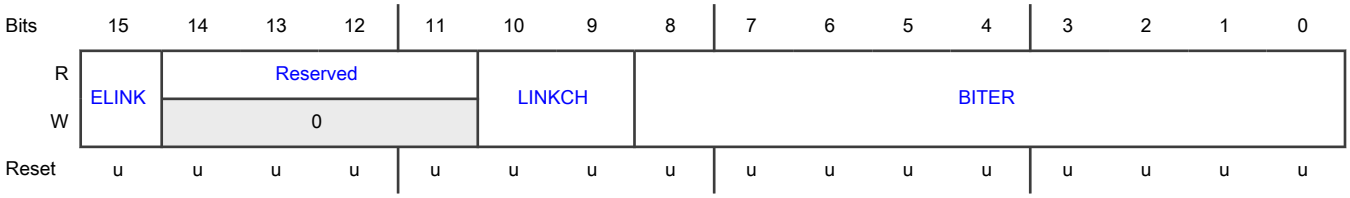
Function

If the TCDn_BITER[ELINK] field is set, the TCDn_BITER register is defined as follows.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram



Fields

Field	Function
15 ELINK	Enable Link As the channel completes the minor loop, this flag enables linking to another channel as defined by BITER[LINKCH]. The link target channel initiates a channel service request via an internal mechanism that sets the TCDn_CSR[START] field of the specified channel. If channel linking disables, the BITER value extends to 15 bits in place of a link channel number. If the major loop is exhausted, this link mechanism is suppressed in favor of the MAJORELINK channel linking. **NOTE** When the software loads the TCD, this field must be set equal to the corresponding CITER field; otherwise, a configuration error is reported. As the major iteration count is exhausted, eDMA reloads the contents of this field into the CITER field. 0b - Channel-to-channel linking disabled 1b - Channel-to-channel linking enabled
14-11 —	Reserved
10-9 LINKCH	Link Channel Number If channel-to-channel linking is enabled (ELINK = 1), then after the minor loop is exhausted, the eDMA engine initiates a channel service request at the channel defined by this field by setting that channel's TCDn_CSR[START] field. **NOTE** When the software loads the TCD, this field must be set equal to the corresponding CITER field; otherwise, a configuration error is reported. As the major iteration count is exhausted, eDMA reloads the contents of this field into the CITER field.
8-0 BITER	Starting Major Iteration Count As the transfer control descriptor is first loaded by software, this 9-bit (ELINK = 1) or 15-bit (ELINK = 0) field must be set equal to the value in the CITER field. As the major iteration count is exhausted, eDMA reloads the contents of this field into the CITER field. If the channel is configured to execute a single service request, the initial values of BITER and CITER must be 0x0001.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Chapter 15
Wakeup Unit (WUU)

15.1 Chip-specific WUU information

Table 87. Reference links to related information

Topic	Related module	Reference
Full description	WUU	WUU
System memory map		Memory map
Clocking		Clock distribution
Power Management		Power Management
Signal multiplexing	Port control	Signal multiplexing

15.1.1 Module instances

This chip contains one instance of the WUU module, WUU0.

15.1.2 Clock signal names

LPO clock is clk_16k[0].

15.1.3 WUU sources

The chip uses the following inputs as wakeup sources to the WUU module. The WUU_Px inputs are connections to external physical pins. The WUU_MxIF inputs are connections to the internal peripheral interrupt flags from different modules that can operate in low-power modes. The WUU_MxDR inputs are connections to the internal peripheral Asynchronous DMA or module Triggers from different modules that can operate in low-power modes.

NOTE

In addition to the WUU wakeup sources, the device also wakes from low power modes when NMI or RESET pins are enabled and the respective pin is asserted.

Table 88. Wakeup sources for WUU inputs

WUU input	Source description
WUU_P0	Reserved
WUU_P1	Reserved
WUU_P2	P0_16
WUU_P3	Reserved
WUU_P4	Reserved
WUU_P5	Reserved
WUU_P6	P1_0 / LPTMR0_ALT3
WUU_P7	P1_3
WUU_P8	Reserved

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table 88. Wakeup sources for WUU inputs (continued)

WUU input	Source description
WUU_P9	Reserved
WUU_P10	P1_8
WUU_P11	P1_11
WUU_P12	Reserved
WUU_P13	Reserved
WUU_P14	Reserved
WUU_P15	Reserved
WUU_P16	Reserved
WUU_P17	Reserved
WUU_P18	P2_0
WUU_P19	P2_3
WUU_P20	P2_12
WUU_P21	Reserved
WUU_P22	P3_0
WUU_P23	P3_8
WUU_P24	P3_11
WUU_P25	P3_14
WUU_P26	P3_28
WUU_P27	P3_29
WUU_P28	Reserved
WUU_P29	Reserved
WUU_P30	Reserved
WUU_P31	CMP0_OUT, CMP0 hard output
WUU_M0IF	SPC interrupt
WUU_M1IF	RTC
WUU_M2IF	Reserved
WUU_M3IF	Reserved
WUU_M4IF	Reserved
WUU_M5IF	Reserved
WUU_M6IF	LPTMR0 Interrupt
WUU_M7IF	Reserved
WUU_M8IF	Reserved

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table 88. Wakeup sources for WUU inputs (continued)

WUU input	Source description
WUU_M9IF	Reserved
WUU_M0DR	Reserved
WUU_M1DR	Reserved
WUU_M2DR	Reserved
WUU_M3DR	Reserved
WUU_M4DR	LPTMR0 DMA request
WUU_M5DR	Reserved
WUU_M6DR	LPTMR0 trig
WUU_M7DR	Reserved
WUU_M8DR	Reserved
WUU_M9DR	Reserved

15.1.4 Access modes

- If the privilege access is disabled (SYSCON.PROTLVL[PRIV] = 0b), this peripheral is accessible in the user mode.
- If the privilege access is enabled (SYSCON.PROTLVL[PRIV] = 1b), this peripheral is accessible only in the privilege mode.

15.2 Overview

WUU facilitates in selecting external pins and on-chip modules as interrupt wake-up sources from Power Down/Deep Power Down power modes. You can also configure the external pins to act as interrupt wake-up sources in all power modes. WUU helps pins and modules generate a temporary wake-up from Power Down mode for servicing DMA or trigger events. Digital filtering is also available for the external wake-up pins.

See the chip-specific WUU information section for input sources to WUU.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

15.2.1 Block diagram

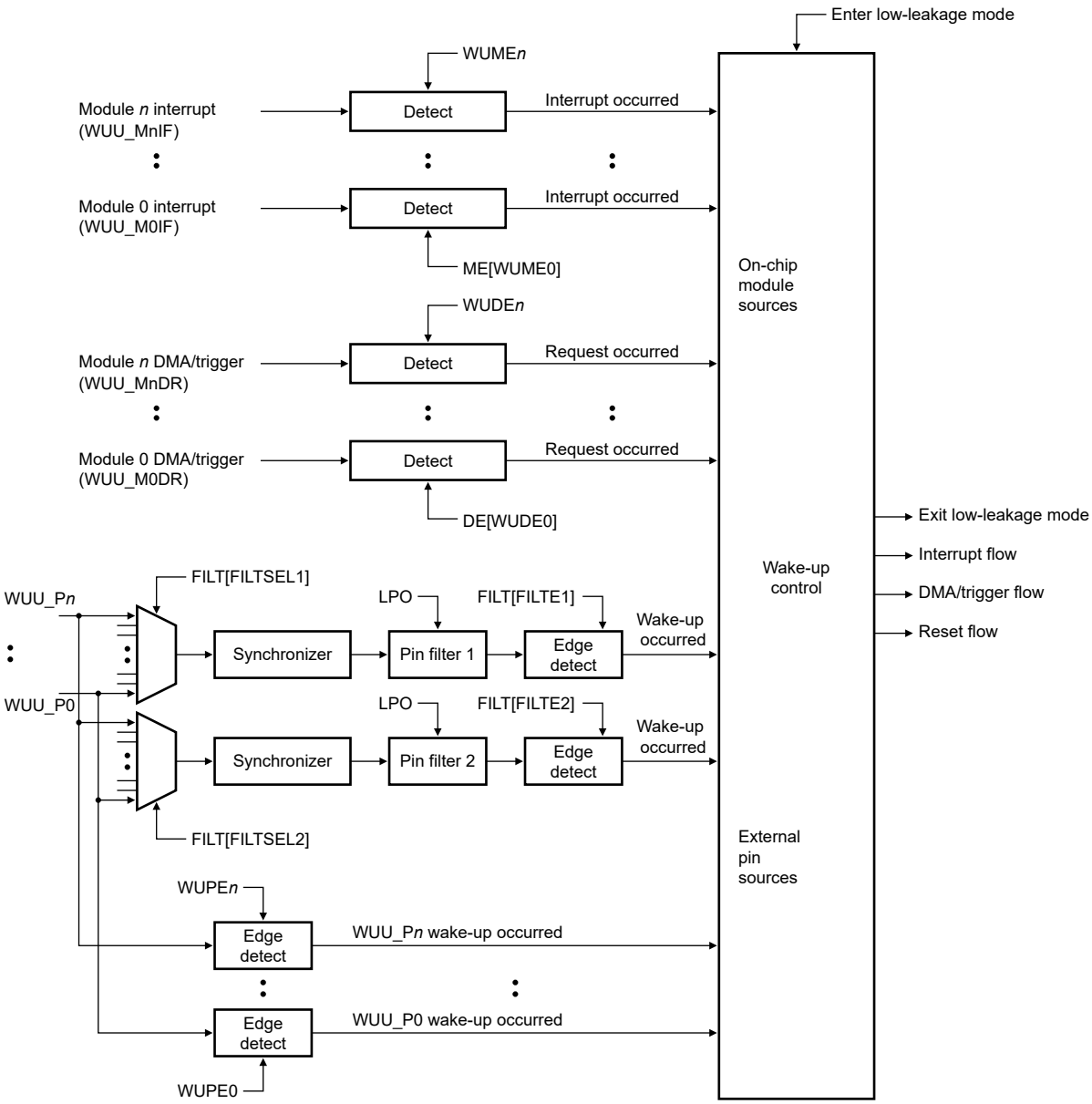


Figure 24. WUU block diagram

15.2.2 Features

- Supports external input pins and on-chip modules with individual enable bits to wake the chip from low-leakage modes.
- Contains input sources that may be external pins or from on-chip modules capable of running in Power Down or Deep Power Down power modes. See the chip-specific WUU information for the wake-up input sources for this chip.
- Supports configuring on-chip modules as DMA or trigger sources for temporary wake-up from Power Down mode.
- Supports configuring external pins as DMA request or trigger sources for temporary wake-up from Power Down mode.
- Provides external input pins programmable for falling-edge, rising-edge, or any edge detection.
- Provides optional digital filters to qualify an external pin detect. Disabling the LPO clock bypasses and disables filters.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

- Filters all external input pins; but only 2 filters are available at a given time.
- Enables external input pin and filter detection in all power modes (not limited to Power Down/Deep Power Down power modes).

15.3 Functional description

15.3.1 Operation

WUU allows on-chip modules and external input pins as a source of wake-up from low-leakage modes.

WUU contains pin enables for each external pin and on-chip module. For each external pin, you can disable or select the edge type for the wake-up with the following options:

- Falling edge
- Rising edge
- Either edge

Exit due to a pin event triggers a WUU interrupt after wake-up is complete. When enabling an external pin as a wake-up source, configure the pin as an input pin. You can also configure a pin to generate an interrupt from a Power Down/Deep Power Down mode or a DMA/trigger event for temporary wake-up from Power Down mode. Detection logic for a pin can optionally remain enabled during all power modes using Pin Mode Configuration (PMC).

On the basis of the LPO clock, WUU implements optional 3-cycle glitch filters. Detected external pin must remain asserted until the enabled glitch filter times out. Additional latency of up to two cycles is due to synchronization, which results in a total of up to five cycles of delay before the detect circuit alerts the system to the wake-up event when the filter function is enabled. Two wake-up detect filters are available for selected external pins. Glitch filtering is not provided for the on-chip modules. You can configure a filter to generate an interrupt from Power Down/Deep Power Down mode or a DMA/trigger event for temporary wake-up from Power Down mode. Detection logic for a filter can optionally remain enabled during all power modes using Pin Filter Mode Configuration (FMC).

For on-chip module interrupts, ME[WUME*n*] enables the associated module interrupt as a wake-up source. Exit due to a module interrupt will not trigger a WUU interrupt.

For on-chip module DMA/trigger events, DE[WUDE*n*] enables the associated module DMA/trigger event as a wake-up source. Exit due to a module DMA/trigger event does not trigger a WUU interrupt.

15.3.2 Modes of operation

WUU becomes functional on entry into a low-leakage power mode. After recovery from Power Down mode, WUU is immediately disabled. After recovery from Deep Power Down mode, WUU continues to detect wake-up events until you have reinitialized the system and deasserted isolation. Detection of external pin/filter events can also optionally remain enabled in all power modes using Pin Mode Configuration (PMC)/Pin Filter Mode Configuration (FMC).

15.3.2.1 Power Down mode

Wake-up events due to either an external pin input (WUU_Px) or an on-chip module interrupt (WUU_MxIF) result in a CPU interrupt flow to begin user code execution. The source of the wake-up determines the subsequent path of code execution:

- Pin events trigger the WUU interrupt service routine.
- Module interrupts trigger that same module's interrupt service routine.

NOTE

The interrupt controller must not mask the WUU interrupt or any on-chip module interrupt to avoid a scenario where the system does not fully recover from Power Down mode after a wake event (e.g., an external pin event).

Wake-up events due to an on-chip module request (WUU_MxDR) DMA/trigger event result in a temporary exit from Power Down mode to allow the request to be serviced when the CPU clock remains gated. The system reenters Power Down mode after servicing the request.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Wake-up events due to an external pin (WUU_Px) DMA/trigger event result in a temporary exit from Power Down mode to allow the request to be serviced when the CPU clock remains gated. The system reenters Power Down mode after servicing the request.

15.3.2.2 Non–low-leakage modes

WUU is inactive in all non–low-leakage modes, with the exception of external pin/filter detection which can explicitly be kept enabled during all power modes using Pin Mode Configuration (PMC)/Pin Filter Mode Configuration (FMC). WUU chips are accessible in non–low-leakage modes and are available for configuring and reading status when bus transactions are possible.

15.3.3 Clocking

During chip low-power operation, in order to provide the wake-up function for the chip, ensure that clocking is available for WUU by the LPO clock. See the chip-specific WUU information for the chip clock connection details.

15.3.4 Reset

WUU resets after VSYS warm reset. This is the global chip reset, but it does not include the reset that asserts due to wake-up from low-power mode.

15.3.5 Interrupts

You can enable on-chip module interrupts as wake-up sources in Module Interrupt Enable (ME).

15.3.6 DMA

You can enable on-chip DMA requests as wake-up sources in Module DMA/Trigger Enable (DE).

With external pin inputs, you can generate a DMA request using either rising edge, falling edge, or both edge detection in Pin DMA/Trigger Configuration 1 (PDC1) and Pin DMA/Trigger Configuration 2 (PDC2).

15.4 External signals

Table 89. External signals

Signal	Description	I/O
WUU_Pn	External pin wake-up inputs; can be enabled to detect a rising edge, a falling edge, or any change.	I
WUU_MnIF	On-chip module interrupt flags	I
WUU_MnDR	On-chip module DMA/trigger events	I

15.5 Initialization

For an enabled module wake-up input, clear the module's flag before entering Power Down or Deep Power Down mode to avoid an immediate exit from the mode.

Clear the flags associated with external input pins, filtered and unfiltered, prior to entry to Power Down or Deep Power Down mode.

When an external wake-up pin filter is first enabled in Pin Filter (FILT), filter operation begins immediately. After enabling an external pin filter or changing its source pin, wait for at least five LPO clock cycles to allow the filter to initialize before entering Power Down or Deep Power Down mode or before enabling filter detection for all power modes in Pin Filter Mode Configuration (FMC).

NOTE

After recovering from a Deep Power Down mode, you must restore the chip configuration before relaxing any power domain isolation controls. In particular, to avoid any WUU flag from being falsely set, restore the pin configuration for the enabled WUU wake-up pins before removing any voltage isolation between power domains.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

15.6 Application information

To enable an external pin as a wake-up source in low-power modes, follow these steps:

1. If the flag is not 0, write 1 to PF[WUF*n*].
2. Configure the PDC1[WUPDC*n*] and PDC2[WUPDC*n*] as either an interrupt or DMA request.
3. Configure the PE1[WUPE*n*] and PE2[WUPE*n*] for rising edge, falling edge, or both edge detection.

When you have configured an external pin to be an active wake-up source in all power modes (PMC[WUPMC*n*] = 1), you must not change the corresponding wake-up pin enable (PE1[WUPE*n*] and PE2[WUPE*n*]) and wake-up pin configuration (PDC1[WUPDC*n*] and PDC2[WUPDC*n*]) settings for that pin. To modify the PE1[WUPE*n*], PE2[WUPE*n*], PDC1[WUPDC*n*] or PDC2[WUPDC*n*] settings, follow these steps:

1. Write 0 to PMC[WUPMC*n*] for that pin.
2. Update the corresponding PE1[WUPE*n*], PE2[WUPE*n*], PDC1[WUPDC*n*] or PDC2[WUPDC*n*] settings as needed.
3. Write 1 to PMC[WUPMC*n*] for that pin.

15.7 Memory map and register definition

This section includes the WUU module memory map and detailed descriptions of all registers.

15.7.1 WUU register descriptions

NOTE

You can write to WUU registers only in Supervisor mode. A bus error results from write accesses in User mode.

VSYS warm reset resets all WUU registers.

15.7.1.1 WUU memory map

WUU0 base address: 4009_2000h

Offset	Register	Width (In bits)	Access	Reset value
0h	Version ID (VERID)	32	R	0100_0001h
4h	Parameter (PARAM)	32	R	2020_2002h
8h	Pin Enable 1 (PE1)	32	RW	0000_0000h
Ch	Pin Enable 2 (PE2)	32	RW	0000_0000h
18h	Module Interrupt Enable (ME)	32	RW	0000_0000h
1Ch	Module DMA/Trigger Enable (DE)	32	RW	0000_0000h
20h	Pin Flag (PF)	32	RW	0000_0000h
30h	Pin Filter (FILT)	32	RW	0000_0000h
38h	Pin DMA/Trigger Configuration 1 (PDC1)	32	RW	0000_0000h
3Ch	Pin DMA/Trigger Configuration 2 (PDC2)	32	RW	0000_0000h
48h	Pin Filter DMA/Trigger Configuration (FDC)	32	RW	0000_0000h
50h	Pin Mode Configuration (PMC)	32	RW	0000_0000h
58h	Pin Filter Mode Configuration (FMC)	32	RW	0000_0000h

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

15.7.1.2 Version ID (VERID)

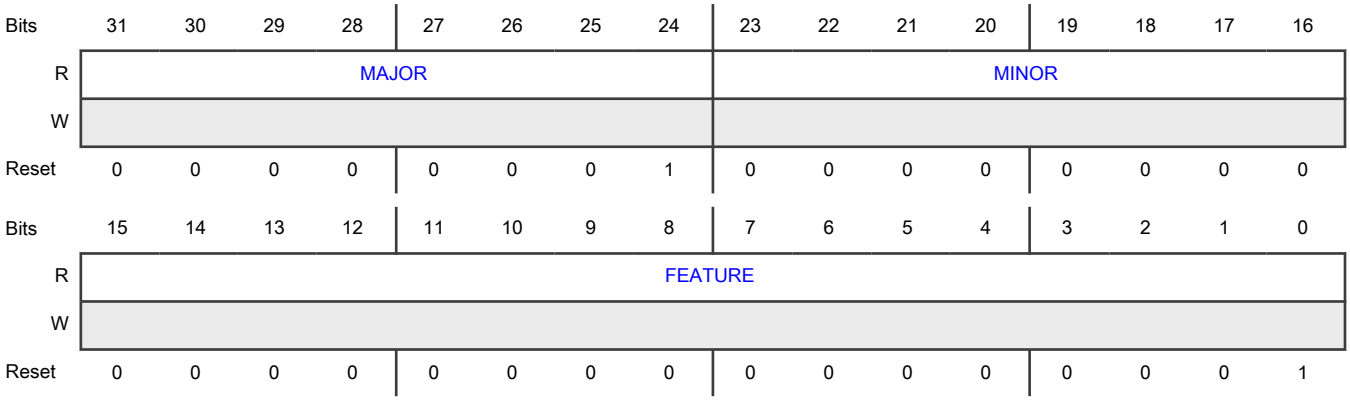
Offset

Register	Offset
VERID	0h

Function

Contains version numbers for the module design and feature set.

Diagram



Fields

Field	Function
31-24 MAJOR	Major Version Number Specifies the major version number for the module specification.
23-16 MINOR	Minor Version Number Specifies the minor version number for the module specification.
15-0 FEATURE	Feature Specification Number Specifies the feature set number. 0000_0000_0000_0000b - Standard features implemented 0000_0000_0000_0001b - Support for DMA/Trigger generation from wake-up pins and filters enabled. Support for external pin/filter detection during all power modes enabled. All other values are reserved.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

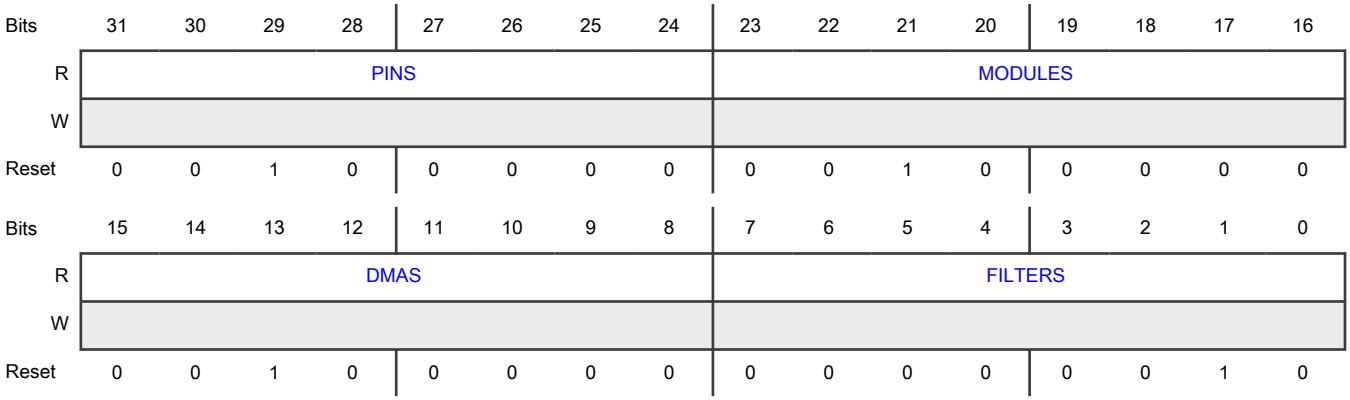
15.7.1.3 Parameter (PARAM)

Offset

Register	Offset
PARAM	4h

Function
Contains parameter values implemented in the module.

Diagram



Fields

Field	Function
31-24 PINS	Pin Number Indicates the number of pin wake-up sources supported.
23-16 MODULES	Module Number Indicates the number of module wake-up sources.
15-8 DMAS	DMA Number Indicates the number of DMA wake-up sources.
7-0 FILTERS	Filter Number Indicates the number of pin filters.

15.7.1.4 Pin Enable 1 (PE1)

Offset

Register	Offset
PE1	8h

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Function

Enables and selects the edge detect type for the available external wake-up input pins in the range from WUU_P0 to WUU_P15.

NOTE

- Do not modify the value of PE1[WUPE*n*] when its corresponding PMC[WUPMC*n*] = 1.
- VSYS warm reset resets this register.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0		0		0		0		WUPE11		WUPE10		0		0	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	WUPE7		WUPE6		0		0		0		WUPE2		0		0	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-30 Reserved15	Reserved
29-28 Reserved14	Reserved
27-26 Reserved13	Reserved
25-24 Reserved12	Reserved
23-22 WUPE11	Wake-up Pin Enable for WUU_P11 Enables the external input pin for wake-up. This field configures the edge detection as follows: • For field value = 01b, when configured as an interrupt/DMA request: Detect on rising edge. When configured as a trigger event: Detect on high level. • For field value = 10b, when configured as an interrupt/DMA request: Detect on falling edge. When configured as a trigger event: Detect on low level. • For field value = 11b, when configured as an interrupt/DMA request: Detect on any edge. 00b - Disable

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	01b - Enable (detect on rising edge or high level) 10b - Enable (detect on falling edge or low level) 11b - Enable (detect on any edge)
21-20 WUPE10	Wake-up Pin Enable for WUU_P10 Enables the external input pin for wake-up. This field configures the edge detection as follows: • For field value = 01b, when configured as an interrupt/DMA request: Detect on rising edge. When configured as a trigger event: Detect on high level. • For field value = 10b, when configured as an interrupt/DMA request: Detect on falling edge. When configured as a trigger event: Detect on low level. • For field value = 11b, when configured as an interrupt/DMA request: Detect on any edge. 00b - Disable 01b - Enable (detect on rising edge or high level) 10b - Enable (detect on falling edge or low level) 11b - Enable (detect on any edge)
19-18 Reserved9	Reserved
17-16 Reserved8	Reserved
15-14 WUPE7	Wake-up Pin Enable for WUU_P7 Enables the external input pin for wake-up. This field configures the edge detection as follows: • For field value = 01b, when configured as an interrupt/DMA request: Detect on rising edge. When configured as a trigger event: Detect on high level. • For field value = 10b, when configured as an interrupt/DMA request: Detect on falling edge. When configured as a trigger event: Detect on low level. • For field value = 11b, when configured as an interrupt/DMA request: Detect on any edge. 00b - Disable 01b - Enable (detect on rising edge or high level) 10b - Enable (detect on falling edge or low level) 11b - Enable (detect on any edge)
13-12 WUPE6	Wake-up Pin Enable for WUU_P6 Enables the external input pin for wake-up. This field configures the edge detection as follows: • For field value = 01b, when configured as an interrupt/DMA request: Detect on rising edge. When configured as a trigger event: Detect on high level.

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	• For field value = 10b, when configured as an interrupt/DMA request: Detect on falling edge. When configured as a trigger event: Detect on low level. • For field value = 11b, when configured as an interrupt/DMA request: Detect on any edge. 00b - Disable 01b - Enable (detect on rising edge or high level) 10b - Enable (detect on falling edge or low level) 11b - Enable (detect on any edge)
11-10 Reserved5	Reserved
9-8 Reserved4	Reserved
7-6 Reserved3	Reserved
5-4 WUPE2	Wake-up Pin Enable for WUU_P2 Enables the external input pin for wake-up. This field configures the edge detection as follows: • For field value = 01b, when configured as an interrupt/DMA request: Detect on rising edge. When configured as a trigger event: Detect on high level. • For field value = 10b, when configured as an interrupt/DMA request: Detect on falling edge. When configured as a trigger event: Detect on low level. • For field value = 11b, when configured as an interrupt/DMA request: Detect on any edge. 00b - Disable 01b - Enable (detect on rising edge or high level) 10b - Enable (detect on falling edge or low level) 11b - Enable (detect on any edge)
3-2 Reserved1	Reserved
1-0 Reserved0	Reserved

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

15.7.1.5 Pin Enable 2 (PE2)

Offset

Register	Offset
PE2	Ch

Function

Contains the field to enable and select the edge detect type for the available external wake-up input pins in the range from WUU_P16 to WUU_P31.

NOTE

- Do not modify the value of PE2[WUPE*n*] when its corresponding PMC[WUPMC*n*] = 1.
- VSYS warm reset resets this register.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	WUPE31		0		0		0		WUPE27		WUPE26		WUPE25		WUPE24	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	WUPE23		WUPE22		0		WUPE20		WUPE19		WUPE18		0		0	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-30 WUPE31	Wake-up Pin Enable for WUU_P31 Enables the external input pin for wake-up and configures the edge detection. • For field value = 01b, when configured as an interrupt/DMA request: Detect on rising edge. When configured as a trigger event: Detect on high level. • For field value = 10b, when configured as an interrupt/DMA request: Detect on falling edge. When configured as a trigger event: Detect on low level. • For field value = 11b, when configured as an interrupt/DMA request: Detect on any edge. 00b - Disable 01b - Enable (detect on rising edge or high level) 10b - Enable (detect on falling edge or low level) 11b - Enable (detect on any edge)

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
29-28 Reserved30	Reserved
27-26 Reserved29	Reserved
25-24 Reserved28	Reserved
23-22 WUPE27	Wake-up Pin Enable for WUU_P27 Enables the external input pin for wake-up and configures the edge detection. • For field value = 01b, when configured as an interrupt/DMA request: Detect on rising edge. When configured as a trigger event: Detect on high level. • For field value = 10b, when configured as an interrupt/DMA request: Detect on falling edge. When configured as a trigger event: Detect on low level. • For field value = 11b, when configured as an interrupt/DMA request: Detect on any edge. 00b - Disable 01b - Enable (detect on rising edge or high level) 10b - Enable (detect on falling edge or low level) 11b - Enable (detect on any edge)
21-20 WUPE26	Wake-up Pin Enable for WUU_P26 Enables the external input pin for wake-up and configures the edge detection. • For field value = 01b, when configured as an interrupt/DMA request: Detect on rising edge. When configured as a trigger event: Detect on high level. • For field value = 10b, when configured as an interrupt/DMA request: Detect on falling edge. When configured as a trigger event: Detect on low level. • For field value = 11b, when configured as an interrupt/DMA request: Detect on any edge. 00b - Disable 01b - Enable (detect on rising edge or high level) 10b - Enable (detect on falling edge or low level) 11b - Enable (detect on any edge)
19-18 WUPE25	Wake-up Pin Enable for WUU_P25 Enables the external input pin for wake-up and configures the edge detection. • For field value = 01b, when configured as an interrupt/DMA request: Detect on rising edge. When configured as a trigger event: Detect on high level.

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	• For field value = 10b, when configured as an interrupt/DMA request: Detect on falling edge. When configured as a trigger event: Detect on low level. • For field value = 11b, when configured as an interrupt/DMA request: Detect on any edge. 00b - Disable 01b - Enable (detect on rising edge or high level) 10b - Enable (detect on falling edge or low level) 11b - Enable (detect on any edge)
17-16 WUPE24	Wake-up Pin Enable for WUU_P24 Enables the external input pin for wake-up and configures the edge detection. • For field value = 01b, when configured as an interrupt/DMA request: Detect on rising edge. When configured as a trigger event: Detect on high level. • For field value = 10b, when configured as an interrupt/DMA request: Detect on falling edge. When configured as a trigger event: Detect on low level. • For field value = 11b, when configured as an interrupt/DMA request: Detect on any edge. 00b - Disable 01b - Enable (detect on rising edge or high level) 10b - Enable (detect on falling edge or low level) 11b - Enable (detect on any edge)
15-14 WUPE23	Wake-up Pin Enable for WUU_P23 Enables the external input pin for wake-up and configures the edge detection. • For field value = 01b, when configured as an interrupt/DMA request: Detect on rising edge. When configured as a trigger event: Detect on high level. • For field value = 10b, when configured as an interrupt/DMA request: Detect on falling edge. When configured as a trigger event: Detect on low level. • For field value = 11b, when configured as an interrupt/DMA request: Detect on any edge. 00b - Disable 01b - Enable (detect on rising edge or high level) 10b - Enable (detect on falling edge or low level) 11b - Enable (detect on any edge)
13-12 WUPE22	Wake-up Pin Enable for WUU_P22 Enables the external input pin for wake-up and configures the edge detection. • For field value = 01b, when configured as an interrupt/DMA request: Detect on rising edge. When configured as a trigger event: Detect on high level.

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	• For field value = 10b, when configured as an interrupt/DMA request: Detect on falling edge. When configured as a trigger event: Detect on low level. • For field value = 11b, when configured as an interrupt/DMA request: Detect on any edge. 00b - Disable 01b - Enable (detect on rising edge or high level) 10b - Enable (detect on falling edge or low level) 11b - Enable (detect on any edge)
11-10 Reserved21	Reserved
9-8 WUPE20	Wake-up Pin Enable for WUU_P20 Enables the external input pin for wake-up and configures the edge detection. • For field value = 01b, when configured as an interrupt/DMA request: Detect on rising edge. When configured as a trigger event: Detect on high level. • For field value = 10b, when configured as an interrupt/DMA request: Detect on falling edge. When configured as a trigger event: Detect on low level. • For field value = 11b, when configured as an interrupt/DMA request: Detect on any edge. 00b - Disable 01b - Enable (detect on rising edge or high level) 10b - Enable (detect on falling edge or low level) 11b - Enable (detect on any edge)
7-6 WUPE19	Wake-up Pin Enable for WUU_P19 Enables the external input pin for wake-up and configures the edge detection. • For field value = 01b, when configured as an interrupt/DMA request: Detect on rising edge. When configured as a trigger event: Detect on high level. • For field value = 10b, when configured as an interrupt/DMA request: Detect on falling edge. When configured as a trigger event: Detect on low level. • For field value = 11b, when configured as an interrupt/DMA request: Detect on any edge. 00b - Disable 01b - Enable (detect on rising edge or high level) 10b - Enable (detect on falling edge or low level) 11b - Enable (detect on any edge)
5-4 WUPE18	Wake-up Pin Enable for WUU_P18 Enables the external input pin for wake-up and configures the edge detection.

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	• For field value = 01b, when configured as an interrupt/DMA request: Detect on rising edge. When configured as a trigger event: Detect on high level. • For field value = 10b, when configured as an interrupt/DMA request: Detect on falling edge. When configured as a trigger event: Detect on low level. • For field value = 11b, when configured as an interrupt/DMA request: Detect on any edge. 00b - Disable 01b - Enable (detect on rising edge or high level) 10b - Enable (detect on falling edge or low level) 11b - Enable (detect on any edge)
3-2 Reserved17	Reserved
1-0 Reserved16	Reserved

15.7.1.6 Module Interrupt Enable (ME)

Offset

Register	Offset
ME	18h

Function

Contains the bits to enable an on-chip module interrupt as a wake-up source.

NOTE

VSYS warm reset resets this register.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0	0	WUME6	0	0	0	0	WUME1	WUME0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Fields

Field	Function
31 —	Reserved
30 —	Reserved
29 —	Reserved
28 —	Reserved
27 —	Reserved
26 —	Reserved
25 —	Reserved
24 —	Reserved
23 —	Reserved
22 —	Reserved
21 —	Reserved
20 —	Reserved
19 —	Reserved
18 —	Reserved
17	Reserved

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
—	
16 —	Reserved
15 —	Reserved
14 —	Reserved
13 —	Reserved
12 —	Reserved
11 —	Reserved
10 —	Reserved
9 —	Reserved
8 —	Reserved
7 —	Reserved
6 WUME6	Module Interrupt Wake-up Enable for Module 6 Enables an on-chip module interrupt as a wake-up source input. 0b - Disable 1b - Enable
5 —	Reserved
4 —	Reserved

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
3 —	Reserved
2 —	Reserved
1 WUME1	Module Interrupt Wake-up Enable for Module 1 Enables an on-chip module interrupt as a wake-up source input. 0b - Disable 1b - Enable
0 WUME0	Module Interrupt Wake-up Enable for Module 0 Enables an on-chip module interrupt as a wake-up source input. 0b - Disable 1b - Enable

15.7.1.7 Module DMA/Trigger Enable (DE)

Offset

Register	Offset
DE	1Ch

Function

Contains the bits to enable an on-chip module DMA/trigger event as a wake-up source.

NOTE

VSYS warm reset resets this register.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0	0	WUDE6	0	WUDE4	0	0	0	0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Fields

Field	Function
31 —	Reserved
30 —	Reserved
29 —	Reserved
28 —	Reserved
27 —	Reserved
26 —	Reserved
25 —	Reserved
24 —	Reserved
23 —	Reserved
22 —	Reserved
21 —	Reserved
20 —	Reserved
19 —	Reserved
18 —	Reserved
17	Reserved

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
—	
16 —	Reserved
15 —	Reserved
14 —	Reserved
13 —	Reserved
12 —	Reserved
11 —	Reserved
10 —	Reserved
9 —	Reserved
8 —	Reserved
7 —	Reserved
6 WUDE6	DMA/Trigger Wake-up Enable for Module 6 Enables an on-chip module DMA/trigger event as a wake-up source. 0b - Disable 1b - Enable
5 —	Reserved
4 WUDE4	DMA/Trigger Wake-up Enable for Module 4 Enables an on-chip module DMA/trigger event as a wake-up source.

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	0b - Disable 1b - Enable
3 —	Reserved
2 —	Reserved
1 —	Reserved
0 —	Reserved

15.7.1.8 Pin Flag (PF)

Offset

Register	Offset
PF	20h

Function

Contains the wake-up flags indicating which wake-up source caused the chip to exit Power Down (if the corresponding PMC[WUPMC*n*] = 1) or Deep Power Down mode. For Power Down mode, this is the source causing the CPU interrupt flow. For Deep Power Down mode, this is the source causing the chip reset flow.

To clear a flag, write a 1 to the corresponding PF[WUF*n*]. If set, the wake-up flag (WUF*n*) remains set even if the associated pin wake-up is disabled in its PE*m*[WUPE*n*] field.

NOTE

VSYS warm reset resets this register.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	WUF31	0	0	0	WUF27	WUF26	WUF25	WUF24	WUF23	WUF22	0	WUF20	WUF19	WUF18	0	0
W	W1C				W1C	W1C	W1C	W1C	W1C	W1C		W1C	W1C	W1C		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	WUF11	WUF10	0	0	WUF7	WUF6	0	0	0	WUF2	0	0
W					W1C	W1C			W1C	W1C				W1C		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31 WUF31	Wake-up Flag for WUU_P31 Indicates that an enabled external wake-up pin was a source of exiting a low-leakage power mode. 0b - No 1b - Yes
30 Reserved30	Reserved
29 Reserved29	Reserved
28 Reserved28	Reserved
27 WUF27	Wake-up Flag for WUU_P27 Indicates that an enabled external wake-up pin was a source of exiting a low-leakage power mode. 0b - No 1b - Yes
26 WUF26	Wake-up Flag for WUU_P26 Indicates that an enabled external wake-up pin was a source of exiting a low-leakage power mode. 0b - No 1b - Yes
25 WUF25	Wake-up Flag for WUU_P25 Indicates that an enabled external wake-up pin was a source of exiting a low-leakage power mode.

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	0b - No 1b - Yes
24 WUF24	Wake-up Flag for WUU_P24 Indicates that an enabled external wake-up pin was a source of exiting a low-leakage power mode. 0b - No 1b - Yes
23 WUF23	Wake-up Flag for WUU_P23 Indicates that an enabled external wake-up pin was a source of exiting a low-leakage power mode. 0b - No 1b - Yes
22 WUF22	Wake-up Flag for WUU_P22 Indicates that an enabled external wake-up pin was a source of exiting a low-leakage power mode. 0b - No 1b - Yes
21 Reserved21	Reserved
20 WUF20	Wake-up Flag for WUU_P20 Indicates that an enabled external wake-up pin was a source of exiting a low-leakage power mode. 0b - No 1b - Yes
19 WUF19	Wake-up Flag for WUU_P19 Indicates that an enabled external wake-up pin was a source of exiting a low-leakage power mode. 0b - No 1b - Yes
18 WUF18	Wake-up Flag for WUU_P18 Indicates that an enabled external wake-up pin was a source of exiting a low-leakage power mode. 0b - No 1b - Yes
17 Reserved17	Reserved

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
16 Reserved16	Reserved
15 Reserved15	Reserved
14 Reserved14	Reserved
13 Reserved13	Reserved
12 Reserved12	Reserved
11 WUF11	Wake-up Flag for WUU_P11 Indicates that an enabled external wake-up pin was a source of exiting a low-leakage power mode. 0b - No 1b - Yes
10 WUF10	Wake-up Flag for WUU_P10 Indicates that an enabled external wake-up pin was a source of exiting a low-leakage power mode. 0b - No 1b - Yes
9 Reserved9	Reserved
8 Reserved8	Reserved
7 WUF7	Wake-up Flag for WUU_P7 Indicates that an enabled external wake-up pin was a source of exiting a low-leakage power mode. 0b - No 1b - Yes
6 WUF6	Wake-up Flag for WUU_P6 Indicates that an enabled external wake-up pin was a source of exiting a low-leakage power mode. 0b - No 1b - Yes

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
5 Reserved5	Reserved
4 Reserved4	Reserved
3 Reserved3	Reserved
2 WUF2	Wake-up Flag for WUU_P2 Indicates that an enabled external wake-up pin was a source of exiting a low-leakage power mode. 0b - No 1b - Yes
1 Reserved1	Reserved
0 Reserved0	Reserved

15.7.1.9 Pin Filter (FILT)

Offset

Register	Offset
FILT	30h

Function

Contains control and status fields to enable and configure the digital filter features for an external wake-up pin.

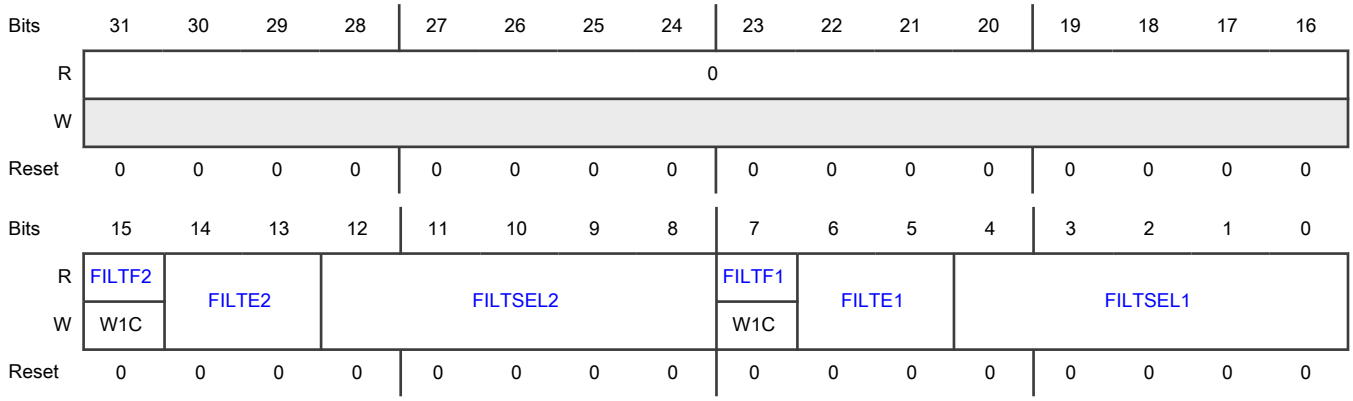
NOTE

VSYS warm reset resets this register.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram



Fields

Field	Function
31-16 —	Reserved
15 FILTF2	Filter 2 Flag Indicates that the filtered external pin was a source of exiting a low-leakage power mode. 0b - No 1b - Yes
14-13 FILTE2	Filter 2 Enable Enables the filter for wake-up. This field configures the edge detection as follows: • For field value = 01b, when configured as an interrupt/DMA request: Detect on rising edge. When configured as a trigger event: Detect on high level • For field value = 10b, when configured as an interrupt/DMA request: Detect on falling edge. When configured as a trigger event: Detect on low level • For field value = 11b, when configured as an interrupt/DMA request: Detect on any edge 00b - Disable 01b - Enable (Detect on rising edge or high level) 10b - Enable (Detect on falling edge or low level) 11b - Enable (Detect on any edge)
12-8 FILTSEL2	Filter 2 Pin Select Selects and filters the external pin WUU_Pn, where *n* equals the value programmed into FILTSEL2.
7 FILTF1	Filter 1 Flag Indicates that the filtered external pin was a source of exiting a low-leakage power mode. 0b - No

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	1b - Yes
6-5 FILTE1	Filter 1 Enable Enables the filter for wake-up. This field configures the edge detection as follows: • For field value = 01b, when configured as an interrupt/DMA request: Detect on rising edge. When configured as a trigger event: Detect on high level • For field value = 10b, when configured as an interrupt/DMA request: Detect on falling edge. When configured as a trigger event: Detect on low level • For field value = 11b, when configured as an interrupt/DMA request: Detect on any edge 00b - Disable 01b - Enable (Detect on rising edge or high level) 10b - Enable (Detect on falling edge or low level) 11b - Enable (Detect on any edge)
4-0 FILTSEL1	Filter 1 Pin Select Selects and filters the external pin WUU_Pn, where *n* equals the value programmed into FILTSEL1.

15.7.1.10 Pin DMA/Trigger Configuration 1 (PDC1)

Offset

Register	Offset
PDC1	38h

Function

Configures the available external wake-up input pins in the range from WUU_P0 to WUU_P15 to generate an interrupt, DMA, or a trigger event when detected.

- When configured as an interrupt, the interrupt is asserted when the edge programmed in Pin Enable 1 (PE1) is detected and remains asserted until the corresponding flag is cleared in Pin Flag (PF).
- When configured as a DMA request, the request is asserted when the edge programmed in Pin Enable 1 (PE1) is detected and remains asserted until either the corresponding flag is cleared in Pin Flag (PF) or until the requested DMA transfer is complete.
- When configured as a trigger event, the trigger is asserted when the level programmed in Pin Enable 1 (PE1) is detected and remains asserted when the input pin is asserted. The corresponding flag in Pin Flag (PF) will not be set.

NOTE

- Do not modify the value of PDC*m*[WUPDC*n*] field when its corresponding PMC[WUPMC*n*] = 1.
- VSYS warm reset resets this register.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0		0		0		0		WUPDC11		WUPDC10		0		0	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	WUPDC7		WUPDC6		0		0		0		WUPDC2		0		0	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-30 Reserved15	Reserved
29-28 Reserved14	Reserved
27-26 Reserved13	Reserved
25-24 Reserved12	Reserved
23-22 WUPDC11	Wake-up Pin Configuration for WUU_P11 Configures an external pin as an interrupt, DMA, or Trigger event. 00b - Interrupt 01b - DMA request 10b - Trigger event 11b - Reserved
21-20 WUPDC10	Wake-up Pin Configuration for WUU_P10 Configures an external pin as an interrupt, DMA, or Trigger event. 00b - Interrupt 01b - DMA request 10b - Trigger event 11b - Reserved
19-18	Reserved

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
Reserved9	
17-16 Reserved8	Reserved
15-14 WUPDC7	Wake-up Pin Configuration for WUU_P7 Configures an external pin as an interrupt, DMA, or Trigger event. 00b - Interrupt 01b - DMA request 10b - Trigger event 11b - Reserved
13-12 WUPDC6	Wake-up Pin Configuration for WUU_P6 Configures an external pin as an interrupt, DMA, or Trigger event. 00b - Interrupt 01b - DMA request 10b - Trigger event 11b - Reserved
11-10 Reserved5	Reserved
9-8 Reserved4	Reserved
7-6 Reserved3	Reserved
5-4 WUPDC2	Wake-up Pin Configuration for WUU_P2 Configures an external pin as an interrupt, DMA, or Trigger event. 00b - Interrupt 01b - DMA request 10b - Trigger event 11b - Reserved
3-2 Reserved1	Reserved
1-0	Reserved

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
Reserved0	

15.7.1.11 Pin DMA/Trigger Configuration 2 (PDC2)

Offset

Register	Offset
PDC2	3Ch

Function

Configures the available external wake-up input pins in the range from WUU_P16 to WUU_P31 to generate an interrupt, DMA, or a trigger event when detected.

- When configured as an interrupt, the interrupt is asserted when the edge programmed in Pin Enable 2 (PE2) is detected and remains asserted until the corresponding flag is cleared in Pin Flag (PF).
- When configured as a DMA request, the request is asserted when the edge programmed in Pin Enable 2 (PE2) is detected and remains asserted until either the corresponding flag is cleared in Pin Flag (PF) or until the requested DMA transfer is complete.
- When configured as a trigger event, the trigger is asserted when the level programmed in Pin Enable 2 (PE2) is detected and remains asserted when the input pin is asserted. The corresponding flag in Pin Flag (PF) will not be set.

NOTE

- Do not modify the value of PDC*m*[WUPDC*n*] field when its corresponding PMC[WUPMC*n*] = 1.
- VSYS warm reset resets this register.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	WUPDC31		0		0		0		WUPDC27		WUPDC26		WUPDC25		WUPDC24	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	WUPDC23		WUPDC22		0		WUPDC20		WUPDC19		WUPDC18		0		0	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Fields

Field	Function
31-30 WUPDC31	Wake-up Pin Configuration for WUU_P31 Configures an external pin as an interrupt, DMA, or Trigger event. 00b - Interrupt 01b - DMA request 10b - Trigger event 11b - Reserved
29-28 Reserved30	Reserved
27-26 Reserved29	Reserved
25-24 Reserved28	Reserved
23-22 WUPDC27	Wake-up Pin Configuration for WUU_P27 Configures an external pin as an interrupt, DMA, or Trigger event. 00b - Interrupt 01b - DMA request 10b - Trigger event 11b - Reserved
21-20 WUPDC26	Wake-up Pin Configuration for WUU_P26 Configures an external pin as an interrupt, DMA, or Trigger event. 00b - Interrupt 01b - DMA request 10b - Trigger event 11b - Reserved
19-18 WUPDC25	Wake-up Pin Configuration for WUU_P25 Configures an external pin as an interrupt, DMA, or Trigger event. 00b - Interrupt 01b - DMA request 10b - Trigger event 11b - Reserved
17-16	Wake-up Pin Configuration for WUU_P24

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
WUPDC24	Configures an external pin as an interrupt, DMA, or Trigger event. 00b - Interrupt 01b - DMA request 10b - Trigger event 11b - Reserved
15-14 WUPDC23	Wake-up Pin Configuration for WUU_P23 Configures an external pin as an interrupt, DMA, or Trigger event. 00b - Interrupt 01b - DMA request 10b - Trigger event 11b - Reserved
13-12 WUPDC22	Wake-up Pin Configuration for WUU_P22 Configures an external pin as an interrupt, DMA, or Trigger event. 00b - Interrupt 01b - DMA request 10b - Trigger event 11b - Reserved
11-10 Reserved21	Reserved
9-8 WUPDC20	Wake-up Pin Configuration for WUU_P20 Configures an external pin as an interrupt, DMA, or Trigger event. 00b - Interrupt 01b - DMA request 10b - Trigger event 11b - Reserved
7-6 WUPDC19	Wake-up Pin Configuration for WUU_P19 Configures an external pin as an interrupt, DMA, or Trigger event. 00b - Interrupt 01b - DMA request 10b - Trigger event 11b - Reserved
5-4	Wake-up Pin Configuration for WUU_P18

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
WUPDC18	Configures an external pin as an interrupt, DMA, or Trigger event. 00b - Interrupt 01b - DMA request 10b - Trigger event 11b - Reserved
3-2 Reserved17	Reserved
1-0 Reserved16	Reserved

15.7.1.12 Pin Filter DMA/Trigger Configuration (FDC)

Offset

Register	Offset
FDC	48h

Function

Configures the external pin filters to generate an interrupt, DMA, or a trigger event when detected.

- When configured as an interrupt, the interrupt is asserted when the edge programmed in FILT*m*[FILTE*n*] is detected and remains asserted until the corresponding FILTF*n* flag is cleared.
- When configured as a DMA request, the request is asserted when the edge programmed in FILT*m*[FILTE*n*] is detected and remains asserted until either the corresponding FILTF*n* flag is cleared or until the requested DMA transfer is complete.
- When configured as a trigger event, the trigger is asserted when the level programmed in Pin Filter (FILT) is detected and remains asserted when the input pin is asserted. The corresponding FILTF*n* flag will not be set.

NOTE

VSYS warm reset resets this register.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0												FILTC2		FILTC1	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-4 —	Reserved
3-2: FILTC2 1-0: FILTC1	Filter Configuration for FILTn Configures a filter as an interrupt, DMA, or Trigger event. 00b - Interrupt 01b - DMA request 10b - Trigger event 11b - Reserved

15.7.1.13 Pin Mode Configuration (PMC)

Offset

Register	Offset
PMC	50h

Function

Configures the detection logic for the available external wake-up input pins to remain enabled during all power modes, not just during Power Down or Deep Power Down mode.

NOTE

VSYS warm reset resets this register.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	WUPMC31	0	0	0	WUPMC27	WUPMC26	WUPMC25	WUPMC24	WUPMC23	WUPMC22	0	WUPMC20	WUPMC19	WUPMC18	0	0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	WUPMC11	WUPMC10	0	0	WUPMC7	WUPMC6	0	0	0	WUPMC2	0	0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31 WUPMC31	Wake-up Pin Mode Configuration for WUU_P31 Configures an external wake-up pin to provide active detection during all power modes. 0b - Active only during a low-leakage mode. You can modify the corresponding fields within Pin Enable (PEn) or Pin DMA/Trigger Configuration (PDCn). 1b - Active during all power modes. Do not modify the corresponding fields within Pin Enable (PEn) or Pin DMA/Trigger Configuration (PDCn).
30 Reserved30	Reserved
29 Reserved29	Reserved
28 Reserved28	Reserved
27 WUPMC27	Wake-up Pin Mode Configuration for WUU_P27 Configures an external wake-up pin to provide active detection during all power modes. 0b - Active only during a low-leakage mode. You can modify the corresponding fields within Pin Enable (PEn) or Pin DMA/Trigger Configuration (PDCn). 1b - Active during all power modes. Do not modify the corresponding fields within Pin Enable (PEn) or Pin DMA/Trigger Configuration (PDCn).
26 WUPMC26	Wake-up Pin Mode Configuration for WUU_P26 Configures an external wake-up pin to provide active detection during all power modes. 0b - Active only during a low-leakage mode. You can modify the corresponding fields within Pin Enable (PEn) or Pin DMA/Trigger Configuration (PDCn).

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	1b - Active during all power modes. Do not modify the corresponding fields within Pin Enable (PEn) or Pin DMA/Trigger Configuration (PDCn).
25 WUPMC25	Wake-up Pin Mode Configuration for WUU_P25 Configures an external wake-up pin to provide active detection during all power modes. 0b - Active only during a low-leakage mode. You can modify the corresponding fields within Pin Enable (PEn) or Pin DMA/Trigger Configuration (PDCn). 1b - Active during all power modes. Do not modify the corresponding fields within Pin Enable (PEn) or Pin DMA/Trigger Configuration (PDCn).
24 WUPMC24	Wake-up Pin Mode Configuration for WUU_P24 Configures an external wake-up pin to provide active detection during all power modes. 0b - Active only during a low-leakage mode. You can modify the corresponding fields within Pin Enable (PEn) or Pin DMA/Trigger Configuration (PDCn). 1b - Active during all power modes. Do not modify the corresponding fields within Pin Enable (PEn) or Pin DMA/Trigger Configuration (PDCn).
23 WUPMC23	Wake-up Pin Mode Configuration for WUU_P23 Configures an external wake-up pin to provide active detection during all power modes. 0b - Active only during a low-leakage mode. You can modify the corresponding fields within Pin Enable (PEn) or Pin DMA/Trigger Configuration (PDCn). 1b - Active during all power modes. Do not modify the corresponding fields within Pin Enable (PEn) or Pin DMA/Trigger Configuration (PDCn).
22 WUPMC22	Wake-up Pin Mode Configuration for WUU_P22 Configures an external wake-up pin to provide active detection during all power modes. 0b - Active only during a low-leakage mode. You can modify the corresponding fields within Pin Enable (PEn) or Pin DMA/Trigger Configuration (PDCn). 1b - Active during all power modes. Do not modify the corresponding fields within Pin Enable (PEn) or Pin DMA/Trigger Configuration (PDCn).
21 Reserved21	Reserved
20 WUPMC20	Wake-up Pin Mode Configuration for WUU_P20 Configures an external wake-up pin to provide active detection during all power modes. 0b - Active only during a low-leakage mode. You can modify the corresponding fields within Pin Enable (PEn) or Pin DMA/Trigger Configuration (PDCn). 1b - Active during all power modes. Do not modify the corresponding fields within Pin Enable (PEn) or Pin DMA/Trigger Configuration (PDCn).
19	Wake-up Pin Mode Configuration for WUU_P19

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
WUPMC19	Configures an external wake-up pin to provide active detection during all power modes. 0b - Active only during a low-leakage mode. You can modify the corresponding fields within Pin Enable (PEn) or Pin DMA/Trigger Configuration (PDCn). 1b - Active during all power modes. Do not modify the corresponding fields within Pin Enable (PEn) or Pin DMA/Trigger Configuration (PDCn).
18 WUPMC18	Wake-up Pin Mode Configuration for WUU_P18 Configures an external wake-up pin to provide active detection during all power modes. 0b - Active only during a low-leakage mode. You can modify the corresponding fields within Pin Enable (PEn) or Pin DMA/Trigger Configuration (PDCn). 1b - Active during all power modes. Do not modify the corresponding fields within Pin Enable (PEn) or Pin DMA/Trigger Configuration (PDCn).
17 Reserved17	Reserved
16 Reserved16	Reserved
15 Reserved15	Reserved
14 Reserved14	Reserved
13 Reserved13	Reserved
12 Reserved12	Reserved
11 WUPMC11	Wake-up Pin Mode Configuration for WUU_P11 Configures an external wake-up pin to provide active detection during all power modes. 0b - Active only during a low-leakage mode. You can modify the corresponding fields within Pin Enable (PEn) or Pin DMA/Trigger Configuration (PDCn). 1b - Active during all power modes. Do not modify the corresponding fields within Pin Enable (PEn) or Pin DMA/Trigger Configuration (PDCn).
10 WUPMC10	Wake-up Pin Mode Configuration for WUU_P10 Configures an external wake-up pin to provide active detection during all power modes. 0b - Active only during a low-leakage mode. You can modify the corresponding fields within Pin Enable (PEn) or Pin DMA/Trigger Configuration (PDCn).

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	1b - Active during all power modes. Do not modify the corresponding fields within Pin Enable (PEn) or Pin DMA/Trigger Configuration (PDCn).
9 Reserved9	Reserved
8 Reserved8	Reserved
7 WUPMC7	Wake-up Pin Mode Configuration for WUU_P7 Configures an external wake-up pin to provide active detection during all power modes. 0b - Active only during a low-leakage mode. You can modify the corresponding fields within Pin Enable (PEn) or Pin DMA/Trigger Configuration (PDCn). 1b - Active during all power modes. Do not modify the corresponding fields within Pin Enable (PEn) or Pin DMA/Trigger Configuration (PDCn).
6 WUPMC6	Wake-up Pin Mode Configuration for WUU_P6 Configures an external wake-up pin to provide active detection during all power modes. 0b - Active only during a low-leakage mode. You can modify the corresponding fields within Pin Enable (PEn) or Pin DMA/Trigger Configuration (PDCn). 1b - Active during all power modes. Do not modify the corresponding fields within Pin Enable (PEn) or Pin DMA/Trigger Configuration (PDCn).
5 Reserved5	Reserved
4 Reserved4	Reserved
3 Reserved3	Reserved
2 WUPMC2	Wake-up Pin Mode Configuration for WUU_P2 Configures an external wake-up pin to provide active detection during all power modes. 0b - Active only during a low-leakage mode. You can modify the corresponding fields within Pin Enable (PEn) or Pin DMA/Trigger Configuration (PDCn). 1b - Active during all power modes. Do not modify the corresponding fields within Pin Enable (PEn) or Pin DMA/Trigger Configuration (PDCn).
1 Reserved1	Reserved

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
0 Reserved0	Reserved

15.7.1.14 Pin Filter Mode Configuration (FMC)

Offset

Register	Offset
FMC	58h

Function

Configures the detection logic for external pin filters to remain enabled during all power modes, not just during Power Down/Deep Power Down mode.

NOTE

VSYS warm reset resets this register.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0														FILTM2	FILTM1
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-2 —	Reserved
1-0 FILTMn	Filter Mode for FILTn Configures an external wake-up pin filter to provide active detection during all power modes. 0b - Active only during Power Down/Deep Power Down mode 1b - Active during all power modes

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Chapter 16
Core Mode Controller (CMC)

16.1 Chip-specific CMC information

Table 90. Reference links to related information

Topic	Related module	Reference
Full description	CMC	CMC
System memory map		Memory map
Clocking		Clock distribution
Power Management		Power Management
Reset		Reset
Signal multiplexing	Port control	Signal multiplexing

16.1.1 Module instances

This chip has one instance of CMC module.

16.2 Overview

CMC provides the sequencing of the CPU and associated logic through the different operating modes.

This chapter describes the available CPU operating modes, including reset, active mode, and low-power modes.

16.2.1 Block diagram

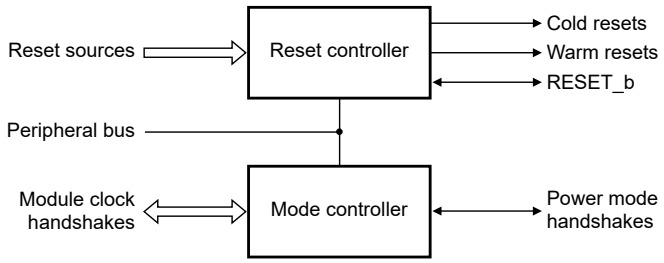


Figure 25. Block diagram

16.2.2 Features

- Configures the low-power mode options
- Provides reset controller including reset status register and warm reset configuration
- Provides boot mode register and status bits

16.3 Functional description

This section provides the functional description of the block.

All information provided in this document is subject to legal disclaimers.
© 2026 NXP B.V. All rights reserved.

16.3.1 Modes of operation

The Arm Cortex-M CPU enters a low-power mode with the execution of a WFI or WFE instruction or via the SLEEPONEXIT mechanism. CMC configures the low-power mode that is entered.

The following table describes the low-power modes available to each core.

Table 91. Operating modes

Mode	Entry	Exit	Description
Reset	POR Warm reset	Active	• Core and peripherals are reset • SRAM optionally retained on power-down wake-up
Active	Reset NVIC WUU	WFI WFE SLEEPONEXIT	Core executing instructions, peripherals optionally active.
Sleep	WFI WFE SLEEPONEXIT	NVIC Reset	• Core waiting for interrupt or event, peripherals optionally active • Core retain state. SRAM is active .
Deep Sleep	WFI WFE SLEEPONEXIT	NVIC Reset	• Core waiting for interrupt or event, peripherals optionally active • Core retain state, peripherals optionally active, SRAM is retained
Power Down	WFI WFE SLEEPONEXIT	WUU via Active Reset	• Core and peripherals waiting for wakeup • Core and peripherals retain state • SRAM is optionally retained
Deep Power Down	WFI WFE SLEEPONEXIT	WUU via Reset Reset	Core and peripherals are not retained. SRAM is optionaly retained

16.3.2 Active mode

This section describes power options in Active mode.

16.3.2.1 Flash memory

This section describes the Flash Memory Low-Power mode.

You can place the internal flash memory in Low-Power mode under the following conditions:

- Under software control (FLASHCR[FLASHDIS])
- When CPU is sleeping (FLASHCR[FLASHDOZE])
- When the chip enters a Deep Sleep mode

When FLASHCR[FLASHDIS] or FLASHCR[FLASHDOZE] becomes 1, you can configure the internal flash memory to automatically exit Low-Power mode during the access on any attempted read access to flash memory space (but not flash peripheral space). FLASHCR[FLASHWAKE] controls this.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

When the internal flash memory enters a low-power mode, the wake-up time of the flash memory automatically delays the first access upon exit from Low-Power mode. If there is no access between the internal flash memory exiting Low-Power and subsequent re-entry of Low-Power mode, the wake-up time of the flash memory delays the re-entry.

16.3.3 Low-power modes

This section describes the low-power modes.

16.3.3.1 Low-power entry

The following steps occur during the Low-Power mode entry sequence.

Table 92. Entry sequence

Step	CKMODE (min)	LPMODE (min)	Description
1	0h	0h	Core enters Low-Power mode by WFI/WFE instruction or via SLEEPONEXIT.
2	1h	0h	• Wake-up interrupt controller is enabled. • Core is clock gated.
3	Fh	1h	• SRAMs enter Low-Power mode (Deep Sleep). • System clocks are gated.
4	Fh	3h	Power domain is placed in Low-Power retention state (Power Down).
5	Fh	Fh	Regulators are powered off (Deep Power-Down).

16.3.3.2 DMA wakeup

You can configure the DMA to generate a wake-up when a DMA request for configured channel asserts. This is supported when you request the DMA to enter Low-Power mode and retain the DMA state (not in Deep Power-Down).

The DMA wakeup triggers an exit from Low-Power mode for everything except the core, which remains clock gated. Re-enter Low-Power mode after the DMA request negates, and the normal low-power mode entry sequence is followed.

Because a DMA wake-up results in everything except the core from exiting Low-Power mode, you must ensure that other modules not involved in the wakeup remain in a known state. To accomplish this, disable the modules before the initial entry in Low-Power mode or configure the Doze field in selected modules.

If the flash memory is not required during the DMA wakeup, NXP recommends that FLASHCR[FLASHDOZE] = 1.

NOTE

If the DMA request does not negate due to the DMA transfer, the chip remains in a higher power state until an interrupt or other wake-up sources can wakeup the core.

An interrupt that occurs during a DMA wakeup causes an immediate exit from Low-Power mode without affecting the DMA transfer.

16.3.3.3 Power domains

CMC configures Low-Power mode of 1 power domain when the core enters a low-power mode with CKMODE = Fh. You can configure the power domain to enter low-power mode. The power domain controlled by the CMC is CORE_MAIN.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Configure all power domains to the same Low-Power mode using Global Power Mode Control (GPMCTRL). Alternatively, configure each PMCTRL register to assign an individual low-power mode to each domain.

16.3.3.4 Debug in low-power modes

When the debugger asserts CDBGPWRUPREQ, it requests the debug logic to power up and enable the functionality depending on DBGCTL[SOD].

- When DBGCTL[SOD] = 0, the core clock remains enabled even when the core is sleeping (CKMODE = 0h). CDBGPWRUPACK remains asserted.
- When DBGCTL[SOD] = 1, ignore the debug request when the core sleeps. CDBPWRUPQACK negates when the core is sleeping.

NOTE

Do not attach a debugger when the JTAG/SWD logic is powered down.

16.3.4 Clocks

- The slow bus clock clocks CMC registers and logic.
- clk_1M clock clocks the timeout counters.

16.3.5 Reset

This section describes the sources of reset and the different resets that you can generate.

16.3.5.1 Reset sources

This section describes the different reset sources.

Power-on reset (POR)

When you initially apply power to the MCU or the supply voltage is below the POR falling threshold, the POR circuit triggers the POR condition.

The POR condition asserts a cold reset in all power domains. SRS[POR] and SRS[VD] become 1 on a POR condition.

Voltage detect (VD)

The voltage detect monitors enable by default and keeps the MCU in reset until the supply voltage rises above the LVD rising threshold. Whenever the voltage detect monitors are enabled, they trigger a reset condition if the supply voltage is outside the voltage detect trip point.

SRS[VD] becomes 1 on voltage detect condition. The voltage detect monitors assert a cold reset in all power domains.

Wakeup (WAKEUP)

On a wakeup from Deep-Power Down mode, the power management logic triggers a Wake-up reset in the power domains that had powered off.

The wake-up reset condition asserts a cold reset in all power domains that were powered down. The system power domain, including the RESET_b pin, is unaffected by a wake-up from Deep-Power Down mode (unless another reset source triggers the wake-up). SRS[WAKEUP] becomes 1 on a wake-up reset condition.

External pin reset (RESET_b)

The RESET_b pin is a bidirectional open-drain pin with an internal pull-up resistor. The RESET_b pin function depends on the mode:

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

- During reset, the RESET_b drives low until the MCU completes initialization, at which point the RESET_b pin is released. If the RESET_b pin asserts externally, then the MCU remains in reset until the RESET_b input is pulled high.
- During active and low-power modes, the RESET_b pin can assert externally to force the MCU into the PIN reset condition.

The RESET_b pin implements a digital filter that you can configure to filter out glitches on the RESET_b pin that are less than 1–32 CMC clock cycles. The filter bypasses in low-power modes when you disable the CMC clock.

The PIN reset condition asserts a warm reset in all power domains. SRS[PIN] and SRS[WARM] become 1 on a PIN reset condition.

Debug access port reset (DAP)

Any debug access port that can initiate a reset request from a connected debugger triggers the DAP reset condition.

The DAP reset condition asserts a warm reset in all power domains. SRS[DAP] and SRS[WARM] become 1 on a DAP reset condition.

Reset timeout (RSTACK)

The reset state machine includes a timeout counter that is triggered by the reset state machine not progressing within 65536 cycles of the clk_1M clock. This triggers the RSTACK reset condition.

The RSTACK reset condition asserts a warm reset in all power domains. SRS[RSTACK], SRS[FATAL], and SRS[WARM] become 1 on an RSTACK reset condition.

Low-power timeout (LPACK)

The low-power entry state machine includes a timeout counter that triggers if a module does not acknowledge entry into Low-Power mode after 65536 cycles of the clk_1M clock. This triggers the LPACK reset condition.

The LPACK reset condition asserts a warm reset in all power domains. SRS[LPACK] and SRS[WARM] become 1 on an LPACK reset condition.

System clock generation (SCG)

The system clock generation includes loss of clock and loss of lock monitors that you can configure to generate a reset, this triggers the SCG reset condition.

The SCG reset condition asserts a warm reset in all power domains. SRS[SCG], SRS[FATAL], and SRS[WARM] become 1 on an SCG reset condition.

Windowed watchdog timer 0 (WWDT0)

The windowed watchdog timer monitors the software by expecting periodic refreshing of the watchdog counter. When this does not occur, it triggers the WWDT0 reset condition.

The WWDT0 reset condition asserts a warm reset in all power domains. SRS[WWDT0] and SRS[WARM] become 1 on a WWDT0 reset condition.

Software (SW)

The software can request a system reset by configuring the system reset request in the core, this triggers the software reset condition.

The software reset condition asserts a warm reset in all power domains. SRS[SW] and SRS[WARM] become 1 on a software reset condition.

Lockup (LOCKUP)

The core enters Lockup state as a result of certain illegal operations, this triggers the LOCKUP reset condition.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

The LOCKUP reset condition asserts an MCU reset in all power domains. SRS[LOCKUP] and SRS[WARM] become 1 on a LOCKUP reset condition.

Windowed watchdog timer 1 (WWDT1)

The windowed watchdog timer monitors the software by expecting periodic refreshing of the watchdog counter. When this does not occur, it triggers the WWDT1 reset condition.

The WWDT1 reset condition asserts a warm reset in all power domains. SRS[WWDT1] and SRS[WARM] become 1 on a WWDT1 reset condition.

JTAG reset (JTAG)

When a JTAG instruction places the chip in reset, it triggers the JTAG reset condition.

The JTAG reset condition asserts a warm reset in all power domains. SRS[JTAG], SRS[FATAL], and SRS[WARM] become 1 on a JTAG reset condition.

16.3.5.2 Reset types

This section describes the different resets that you can generate.

Cold reset

Each power domain generates a cold reset to reset the debug logic and mode control logic. The cold reset for each power domain can assert as a result of the following events.

- Initial POR of the chip
- Voltage detect monitor (LVD, for example)
- Power domain wake-up from Deep-Power Down (varies per domain)

A cold reset does not guarantee the contents of on-chip SRAM.

Warm reset

- Cold Reset or any of the remaining warm reset sources generates a warm reset.
- A warm reset resets most of the logic in each power domain.
- Warm resets are divided into fatal reset sources and non-fatal reset sources.

You can configure the non-fatal reset sources to generate an interrupt instead of the warm reset. The warm reset averts if you can clear the non-fatal reset source (including status flag) within 65536 cycles of the clk_1M clock. Non-fatal resets retain the contents of on-chip SRAM.

You cannot configure the fatal reset sources to generate an interrupt and do not guarantee the contents of on-chip SRAM.

16.3.5.3 Reset sequence

Power-on reset

Perform the following steps as part of the POR (or voltage detect monitor) sequence:

1. RESET_b pin drives low and internal reset signals asserted.
2. POR or LVD signals negate and clocks are enabled in their default configuration.
3. Internal reset remains asserted for 8 CMC clock cycles.
4. Internal reset to flash memory and fuse controllers negates and their initialization sequences commence.
5. Initialization sequences for flash memory and fuse complete.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

6. Internal trim registers are loaded and RESET_b pin is tri-stated.
7. Read the ISP pin status and record it in the Mode (MR0) register.
8. Reset state machine waits for RESET_b pin input to pull high or drive high externally.
9. Internal reset signals negate.
10. Core exits reset and fetches the initial program counter and stack pointer from ROM.

Deep-power down wake-up

Perform the following steps as part of the deep-power down wake-up sequence:

1. Internal reset signals asserted.
2. Wake-up signal negates and clocks are enabled in their default configuration.
3. Internal reset remains asserted for 8 CMC clock cycles.
4. Internal reset to flash memory and fuse controllers negates and commence their initialization sequences.
5. Initialization sequences for flash memory and fuse complete.
6. Internal trim registers are loaded.
7. Internal reset signals negate.
8. Core exits reset and fetches the initial program counter and stack pointer from ROM.

A wake-up from deep-power down via the RESET_b pin follows the warm reset sequence.

Warm reset

Perform the following steps as part of the warm reset sequence:

1. RESET_b pin drives low and internal reset signals asserted.
2. Clocks are enabled in their default configuration.
3. Internal reset remains asserted for 8 CMC clock cycles.
4. Internal reset to flash memory and fuse controllers negates and commence their initialization sequences.
5. Initialization sequences for flash memory and fuse complete.
6. Internal trim registers are loaded and RESET_b pin is tri-stated.
7. Read the ISP pin status and record it in the Mode (MR0) register.
8. Reset state machine waits for RESET_b pin input to pull high or drive high externally.
9. Internal reset signals negate.
10. Core exits reset and fetches the initial program counter and stack pointer from ROM.

16.3.6 Interrupts

CMC generates a single interrupt, which System Reset Interrupt Enable (SRIE) configures.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

16.4 External signals

Table 93. External signals

Signal	Description	Direction
RESET_B	Bidirectional open-drain reset pin with pullup that asserts low during warm reset except when waking up from a low-power mode.	Input or output
ISPMODE_n	The input pin is sampled at the end of the RESET_B assertion and stored in Mode (MR0).	Input

16.5 Initialization

By default, the digital glitch filter is disabled on RESET_b pin. To enable the filter, configure:

- RPC[LPFEN], RPC[FILTEN], or both.
- RPC[FILTCFG].

16.6 Application information

To configure for Deep-Power Down Low-Power mode entry:

1. Write Fh to Clock Control (CKCTRL)
2. Write 8h to Power Mode Protection (PMPROT)
3. Write Fh to Global Power Mode Control (GPMCTRL)
4. Execute WFI instruction

To configure for Power-Down Low-Power mode entry:

1. Write Fh to Clock Control (CKCTRL)
2. Write 2h to Power Mode Protection (PMPROT)
3. Write 3h to Global Power Mode Control (GPMCTRL)
4. Execute WFI instruction

To configure for Sleep mode entry:

1. Write 0h or 1h to Clock Control (CKCTRL)
2. Execute WFI instruction

To configure for Deep Sleep Low-Power mode entry:

1. Write Fh to Clock Control (CKCTRL)
2. Write 1h to Power Mode Protection (PMPROT)
3. Write 1h to Global Power Mode Control (GPMCTRL)
4. Execute WFI instruction

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

16.7 Memory map and register descriptions

This section describes the registers in the CMC module.

16.7.1 CMC register descriptions

NOTE

Different CMC registers reset on different reset types.

NOTE

You must read back the last register written to before executing the WFI instruction. This ensures that before the MCU enters the low power mode, all register writes associated with setting up the low power mode are complete, failure to do this may result in the low power mode not being entered correctly.

16.7.1.1 CMC memory map

CMC base address: 4008_B000h

Offset	Register	Width (In bits)	Access	Reset value
0h	Version ID (VERID)	32	R	0300_0000h
10h	Clock Control (CKCTRL)	32	RW	0000_0000h
14h	Clock Status (CKSTAT)	32	RW	0000_0000h
18h	Power Mode Protection (PMPROT)	32	RW	0000_0000h
1Ch	Global Power Mode Control (GPMCTRL)	32	RW	0000_0000h
20h	Power Mode Control (PMCTRLCORE_MAIN)	32	RW	0000_0000h
80h	System Reset Status (SRS)	32	R	See section
84h	Reset Pin Control (RPC)	32	RW	0000_0000h
88h	Sticky System Reset Status (SSRS)	32	RW	0000_0006h
8Ch	System Reset Interrupt Enable (SRIE)	32	RW	0000_8800h
90h	System Reset Interrupt Flag (SRIF)	32	RW	0000_0000h
A0h	Mode (MR0)	32	RW	0000_0000h
B0h	Force Mode (FM0)	32	RW	0000_0000h
E0h	Flash Control (FLASHCR)	32	RW	0000_0000h
110h	Core Control (CORECTL)	32	RW	0000_0000h
120h	Debug Control (DBGCTL)	32	RW	0000_0000h

16.7.1.2 Version ID (VERID)

Offset

Register	Offset
VERID	0h

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Function

Contains version numbers for module design and feature set.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	MAJOR								MINOR							
W																
Reset	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	FEATURE															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-24 MAJOR	Major Version Number Returns the major version number for the module specification.
23-16 MINOR	Minor Version Number Returns the minor version number for the module specification.
15-0 FEATURE	Feature Specification Number Returns the feature set number.

16.7.1.3 Clock Control (CKCTRL)

Offset

Register	Offset
CKCTRL	10h

Function

Configures the amount of clock gating when the core asserts sleeping due to WFI, WFE, or SLEEPONEXIT.

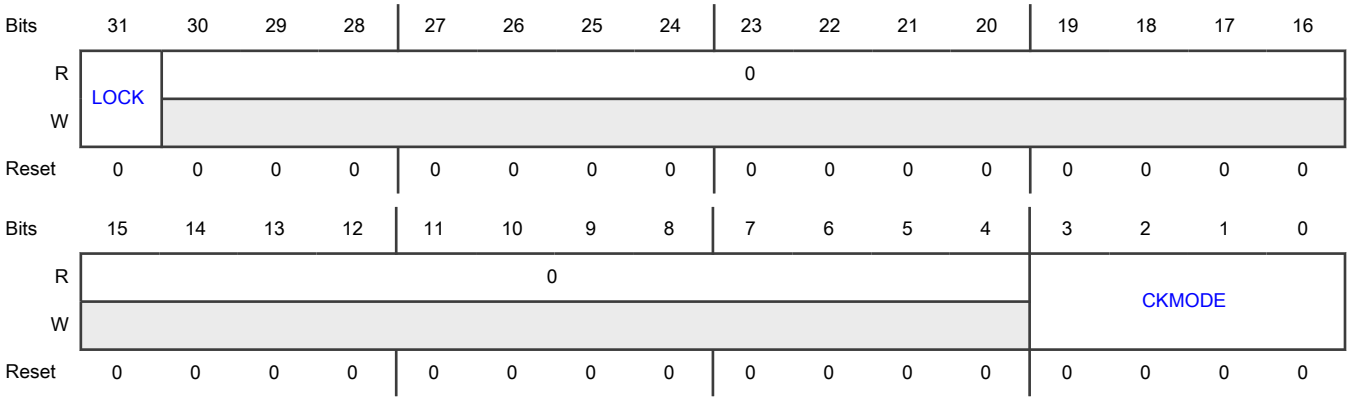
NOTE

This register resets on MAIN warm reset.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram



Fields

Field	Function
31 LOCK	Lock Locks the register and blocks writes until the next reset. 0b - Allowed 1b - Blocked
30-4 —	Reserved
3-0 CKMODE	Clocking Mode Configures the amount of clock gating when the core enters a low-power mode because of WFI, WFE or SLEEPONEXIT. Configuring CKMODE > 0 requires the SLEEPDEEP field in the Arm core to become 1. Configuring PMCTRLx[LPMODE] > 0 requires writing Fh to CKMODE. 0000b - Core clock is on 0001b - Core clock is off 1111b - Core, platform, and peripheral clocks are off, and core enters Low-Power mode

16.7.1.4 Clock Status (CKSTAT)

Offset

Register	Offset
CKSTAT	14h

Function

Returns the clock gating status and wake-up source from the previous Low-Power mode entry, provided the core was clock gated. This requires configuring CKCTRL[CKMODE] > 0 and the wake-up event must occur after the core is clock gated. The register contents are only valid when CKSTAT[VALID] = 1.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

NOTE

This register resets on MAIN warm reset.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	VALID	0														
W	W1C															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	WAKEUP								0				CKMODE			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31 VALID	Clock Status Valid Indicates that the core clock was gated since you last wrote 0 to this field. 0b - Core clock not gated 1b - Core clock was gated due to Low-Power mode entry
30-16 —	Reserved
15-8 WAKEUP	Wake-up Source Returns any wake-up sources from the previous Low-Power mode entry. [0] - Wake-up source is reset interrupt or wakeup from Deep Power-Down [1] - Wake-up source is debug request [2] - Wake-up source is interrupt [3] - Wake-up source is DMA wakeup [4] - Wake-up source is WUU request [5] - Wake-up source is reserved [6] - Wake-up source is reserved [7] - Wake-up source is reserved A wakeup from Deep Power-Down writes 1 to WAKEUP[0]. Depending on Low-Power mode, other wake-up bits may not become 1.
7-4 —	Reserved

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
3-0 CKMODE	Low Power Status Returns the result of the previous Low-Power mode entry. 0000b - Core clock is on 0001b - Core clock is off 1111b - Core, platform, and peripheral clocks are off, and core enters Low-Power mode All other values are reserved.

16.7.1.5 Power Mode Protection (PMPROT)

Offset

Register	Offset
PMPROT	18h

Function

Provides protection for entry in low-power modes, you must allow a low-power mode before configuring any Power Mode Control register (PMCTRLx) to that mode.

NOTE

This register resets on MAIN warm reset.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	LOCK	0														
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0												LPMODE			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31	Lock Register Locks the register and blocks writes until the next reset.

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
LOCK	0b - Allowed 1b - Blocked
30-4 —	Reserved
3-0 LPMODE	Low-Power Mode Bit 0: When set, allows the PMCTRLx[LPMODE] field to be configured for Deep Sleep. Bit 1: When set, allows the PMCTRLx[LPMODE] field to be configured for Power Down. Bit 3: When set, allows the PMCTRLx[LPMODE] field to be configured for Deep Power Down. 0000b - Not allowed 0001b - Allowed 0010b - Allowed 0011b - Allowed 0100b - Allowed 0101b - Allowed 0110b - Allowed 0111b - Allowed 1000b - Allowed 1001b - Allowed 1010b - Allowed 1011b - Allowed 1100b - Allowed 1101b - Allowed 1110b - Allowed 1111b - Allowed

16.7.1.6 Global Power Mode Control (GPMCTRL)

Offset

Register	Offset
GPMCTRL	1Ch

Function

Updates all PMCTRLx registers with the value written.

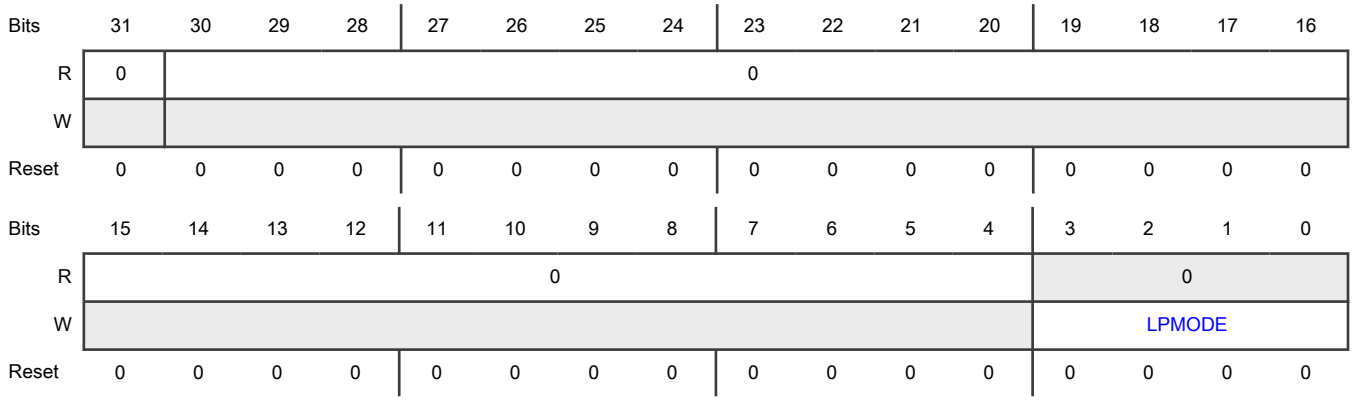
All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

NOTE

This register resets on MAIN cold reset.

Diagram



Fields

Field	Function
31 —	Reserved
30-4 —	Reserved
3-0 LPMODE	Low-Power Mode Specifies that all PMCTRLx[LPMODE] fields update with the value written.

16.7.1.7 Power Mode Control (PMCTRLCORE_MAIN)

Offset

Register	Offset
PMCTRLCORE_MAIN	20h

Function

Configures entry into low-power modes for each power domain, provided that the selected power mode is allowed via an appropriate setting of Power Mode Protection (PMPROT), and CKMODE = Fh.

NOTE

This register resets by MAIN cold reset.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0		0													
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0												LPMODE			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-30 —	Reserved
29-4 —	Reserved
3-0 LPMODE	Low-Power Mode Selects the desired Low-Power mode when a core executes WFI or WFE instruction. Writes to this field are blocked if you have not enabled the protection level using Power Mode Protection (PMPROT). You must not configure this field for the MAIN domain to a lower power mode than any other power domain. **NOTE** See Power domains section for details on allowed combinations. 0000b - Active/Sleep 0001b - Deep Sleep 0011b - Power Down 0111b - Reserved 1111b - Deep-Power Down

16.7.1.8 System Reset Status (SRS)

Offset

Register	Offset
SRS	80h

Function

Updates on every CORE_MAIN warm reset to indicate the type/source of the most recent reset.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	0	0	JTAG	0	0	WWDT1	0	0	0					0	0
W																
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	LOCKUP	SW	WWDT0	SCG	LPACK	RSTACK	DAP	PIN	0		FATAL	WARM	0	VD	POR	WAKEUP
W																
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

Fields

Field	Function
31 —	Reserved
30 —	Reserved
29 —	Reserved
28 JTAG	JTAG System Reset Indicates a JTAG system reset request that causes a reset. This is a fatal reset source and SRAM contents are not guaranteed. 0b - Reset not generated 1b - Reset generated
27 —	Reserved
26 —	Reserved
25 WWDT1	Windowed Watchdog 1 Reset Indicates the WWDT1 timeout that causes a reset. 0b - Reset is not generated 1b - Reset is generated
24	Reserved

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
—	
23 —	Reserved
22-18 —	Reserved
17 —	Reserved
16 —	Reserved
15 LOCKUP	Lockup Reset Indicates the Arm core indication of a LOCKUP event that causes a reset. 0b - Reset not generated 1b - Reset generated
14 SW	Software Reset Indicates a software reset request from the Arm core (SYSRESETREQ) that causes a reset. 0b - Reset not generated 1b - Reset generated
13 WWDT0	Windowed Watchdog 0 Reset Indicates the WWDT 0 timeout that causes a reset. 0b - Reset is not generated 1b - Reset is generated
12 SCG	System Clock Generation Reset Indicates a loss-of-clock or loss-of-lock event in the SCG that causes a reset. 0b - Reset is not generated 1b - Reset is generated
11 LPACK	Low Power Acknowledge Timeout Reset Indicates a timeout in the Low Power Mode entry logic that causes a reset. This timeout is generated if a peripheral does not acknowledge entry into Low-Power mode within 65536 cycles of clk_1M clock. 0b - Reset not generated 1b - Reset generated
10	Reset Timeout

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
RSTACK	Indicates a timeout or other error condition in the system reset generation logic that causes a reset. This is a fatal reset source, and SRAM contents are not guaranteed. 0b - Reset not generated 1b - Reset generated
9 DAP	Debug Access Port Reset Indicates a reset request from a DAP that causes a reset. 0b - Reset was not generated 1b - Reset was generated
8 PIN	Pin Reset Indicates an external assertion of the RESET_b pin that causes a reset. 0b - Reset was not generated 1b - Reset was generated
7-6 —	Reserved
5 FATAL	Fatal Reset Asserts if the last reset source was a fatal reset source. You cannot guarantee SRAM contents following a fatal reset source. 0b - Reset was not generated 1b - Reset was generated
4 WARM	Warm Reset Asserts if the last reset source was a warm reset source. 0b - Reset not generated 1b - Reset generated
3 —	Reserved
2 VD	Voltage Detect Reset Indicates that an enabled voltage detect monitor causes a reset. 0b - Reset not generated 1b - Reset generated
1 POR	Power-on Reset Indicates the POR detection logic that causes a reset.

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	You cannot guarantee SRAM contents following a POR source. 0b - Reset not generated 1b - Reset generated
0 WAKEUP	Wake-up Reset Indicates a wake-up from Deep-Power Down mode that causes a reset. 0b - Reset not generated 1b - Reset generated

16.7.1.9 Reset Pin Control (RPC)

Offset

Register	Offset
RPC	84h

Function

NOTE

MAIN cold reset resets this register.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0						LPFEN	FILTEN	0				FILTCFG			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-10 —	Reserved

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
9 LPFEN	Low-Power Filter Enable Enables the low-power reset pin filter in both Active and Low-Power modes when this field = 1. The RESET_b pin must assert for more than three clk_1M clock cycles to always propagate through the filter. A glitch less than two clk_1M clock cycles never propagates through the filter. 0b - Disables 1b - Enables
8 FILTEN	Filter Enable Enables the slow clock reset pin filter in Active modes. The RESET_b pin must assert for more than FILTCFG + 1 slow system clock cycles to always propagate through the filter. A glitch less than FILTCFG slow system clock cycles never propagates through the filter. If RPC[LPFEN] also becomes 1, then the filters operate in series. 0b - Disables 1b - Enables
7-5 —	Reserved
4-0 FILTCFG	Reset Filter Configuration Configures the reset pin filter's width from 1 to 32 slow system clock cycles.

16.7.1.10 Sticky System Reset Status (SSRS)

Offset

Register	Offset
SSRS	88h

Function

Stores all sources of system reset that has generated a system reset since the last MAIN cold reset and that you have not cleared. SSRS does not update following a core software reset.

NOTE

This register resets on MAIN cold reset.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	0	0	JTAG	0	0	WWDT1	0	0	0					0	0
W				W1C			W1C									
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	LOCKUP	SW	WWDT0	SCG	LPACK	RSTACK	DAP	PIN	0		FATAL	WARM	0	VD	POR	WAKEUP
W	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C			W1C	W1C			W1C	W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0

Fields

Field	Function
31 —	Reserved
30 —	Reserved
29 —	Reserved
28 JTAG	JTAG System Reset Indicates a JTAG system reset that causes a reset. This is a fatal reset source and SRAM contents are not guaranteed. 0b - Reset not generated 1b - Reset generated
27 —	Reserved
26 —	Reserved
25 WWDT1	Windowed Watchdog 1 Reset Indicates the WWDT1 timeout that causes a reset. 0b - Reset is not generated 1b - Reset is generated
24	Reserved

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
—	
23 —	Reserved
22-18 —	Reserved
17 —	Reserved
16 —	Reserved
15 LOCKUP	Lockup Reset Indicates the core lockup that causes a reset. 0b - Reset not generated 1b - Reset generated
14 SW	Software Reset Indicates the software request from the Arm core (SYSRESETREQ) that causes a reset. 0b - Reset not generated 1b - Reset generated
13 WWDT0	Windowed Watchdog 0 Reset Indicates a windowed watchDog timeout that causes a reset. 0b - Reset is not generated 1b - Reset is generated
12 SCG	System Clock Generation Reset Indicates an SCG loss of lock or loss of clock that causes a reset. 0b - Reset is not generated 1b - Reset is generated
11 LPACK	Low Power Acknowledge Timeout Reset Indicates a low power acknowledge timeout that causes a reset. 0b - Reset not generated 1b - Reset generated
10	Reset Timeout

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
RSTACK	Indicates a reset controller timeout that causes a reset. This is a fatal reset source and SRAM contents are not guaranteed. 0b - Reset not generated 1b - Reset generated
9 DAP	DAP Reset Indicates a DAP reset request that causes a reset. 0b - Reset not generated 1b - Reset generated
8 PIN	Pin Reset Indicates a RESET_B pin that causes a reset. 0b - Reset not generated 1b - Reset generated
7-6 —	Reserved
5 FATAL	Fatal Reset Indicates a fatal reset source that causes a reset. 0b - Reset was not generated 1b - Reset was generated
4 WARM	Warm Reset Indicates a warm reset source that causes a reset. 0b - Reset not generated 1b - Reset generated
3 —	Reserved
2 VD	Voltage Detect Reset Indicates that an enabled voltage detect monitor causes a reset. 0b - Reset not generated 1b - Reset generated
1 POR	Power-on Reset Indicates the POR that causes a reset.

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	0b - Reset not generated 1b - Reset generated
0 WAKEUP	Wake-up Reset Indicates the wake-up from Deep-Power Down mode that causes a reset. 0b - Reset not generated 1b - Reset generated

16.7.1.11 System Reset Interrupt Enable (SRIE)

Offset

Register	Offset
SRIE	8Ch

Function

Delays the assertion of a system reset for 65538 cycles of the clk_1M clock when an interrupt is generated. This allows you to perform a graceful shutdown or to abort the warm reset provided you can clear the pending reset source by resetting the source and then clearing the pending flag. This feature cannot delay a cold or fatal warm reset source. System Reset Status (SRS) updates after the warm reset occurs.

The reset interrupt is not supported in Deep-Power Down mode.

NOTE

This register is reset on MAIN Cold Reset.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	0			0	0	WWDT1	0	0	0					0	0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	LOCKUP	SW	WWDT0	SCG	LPACK	0	DAP	PIN	0							
W																
Reset	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Fields

Field	Function
31 —	Reserved
30-28 —	Reserved
27 —	Reserved
26 —	Reserved
25 WWDT1	Windowed Watchdog 1 Reset 0b - Interrupt disabled 1b - Interrupt enabled
24 —	Reserved
23 —	Reserved
22-18 —	Reserved
17 —	Reserved
16 —	Reserved
15 LOCKUP	Lockup Reset **NOTE** A core in the lockup state cannot service interrupts. 0b - Interrupt disabled 1b - Interrupt enabled
14 SW	Software Reset 0b - Interrupt disabled 1b - Interrupt enabled
13	Windowed Watchdog 0 Reset

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
WWDT0	0b - Interrupt disabled 1b - Interrupt enabled
12 SCG	System Clock Generation Reset 0b - Interrupt disabled 1b - Interrupt enabled
11 LPACK	Low Power Acknowledge Timeout Reset 0b - Interrupt disabled 1b - Interrupt enabled
10 —	Reserved
9 DAP	DAP Reset 0b - Interrupt disabled 1b - Interrupt enabled
8 PIN	Pin Reset 0b - Interrupt disabled 1b - Interrupt enabled
7-0 —	Reserved

16.7.1.12 System Reset Interrupt Flag (SRIF)

Offset

Register	Offset
SRIF	90h

Function

Returns the source of the reset interrupt. You can clear the pending reset source by resetting the source and then clearing the pending flag.

NOTE

This register resets on MAIN warm reset.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	0			0	0	WWDT1	0	0	0					0	0
W							W1C									
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	LOCKUP	SW	WWDT0	SCG	LPACK	0	DAP	PIN	0							
W	W1C	W1C	W1C	W1C	W1C		W1C	W1C								
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31 —	Reserved
30-28 —	Reserved
27 —	Reserved
26 —	Reserved
25 WWDT1	Windowed Watchdog 1 Reset 0b - Reset source not pending 1b - Reset source pending
24 —	Reserved
23 —	Reserved
22-18 —	Reserved
17 —	Reserved
16	Reserved

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
—	
15 LOCKUP	Lockup Reset 0b - Reset source not pending 1b - Reset source pending
14 SW	Software Reset 0b - Reset source not pending 1b - Reset source pending
13 WWDT0	Windowed Watchdog 0 Reset 0b - Reset source not pending 1b - Reset source pending
12 SCG	System Clock Generation Reset 0b - Reset source not pending 1b - Reset source pending
11 LPACK	Low Power Acknowledge Timeout Reset 0b - Reset source not pending 1b - Reset source pending
10 —	Reserved
9 DAP	DAP Reset 0b - Reset source not pending 1b - Reset source pending
8 PIN	Pin Reset 0b - Reset source not pending 1b - Reset source pending
7-0 —	Reserved

16.7.1.13 Mode (MR0)

Offset

Register	Offset
MR0	A0h

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Function

Contains the state of the boot mode pins sampled at the end of the reset.

NOTE

This register resets on MAIN warm reset, and the reset value may vary depending on the pin state.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															ISPMODE...
W																W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-1 —	Reserved
0 ISPMODE_n	In System Programming Mode Returns the logic state of the ISPMODE_n pin on the last negation of the RESET_b pin.

16.7.1.14 Force Mode (FM0)

Offset

Register	Offset
FM0	B0h

Function

Contains registers that override the state of the boot mode pins.

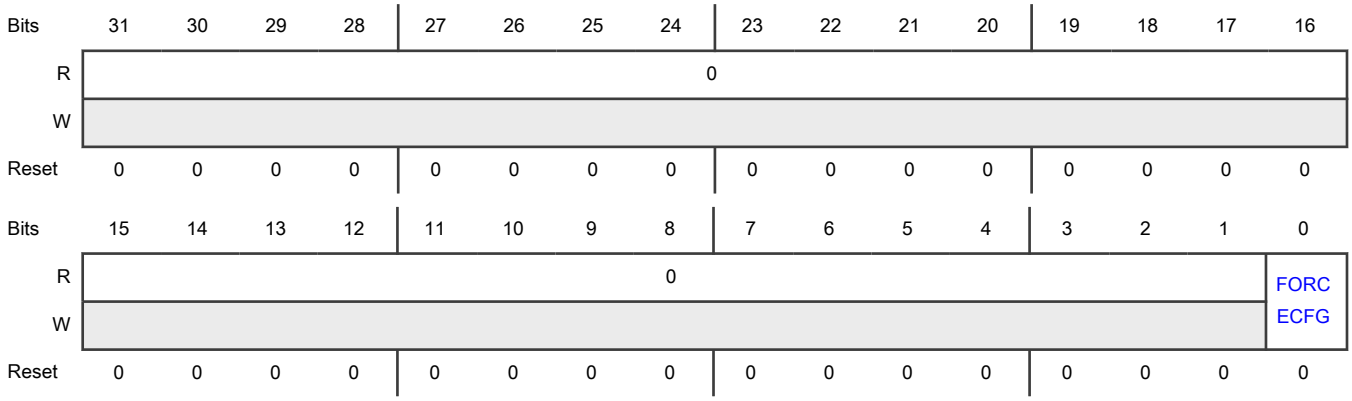
NOTE

This register resets on MAIN cold reset.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram



Fields

Field	Function
31-1 —	Reserved
0 FORCECFG	Boot Configuration Forces the corresponding field in Mode (MR0) to assert on next system reset. 0b - No effect 1b - Asserts

16.7.1.15 Flash Control (FLASHCR)

Offset

Register	Offset
FLASHCR	E0h

Function

Controls Low-Power mode of the on-chip flash memory.

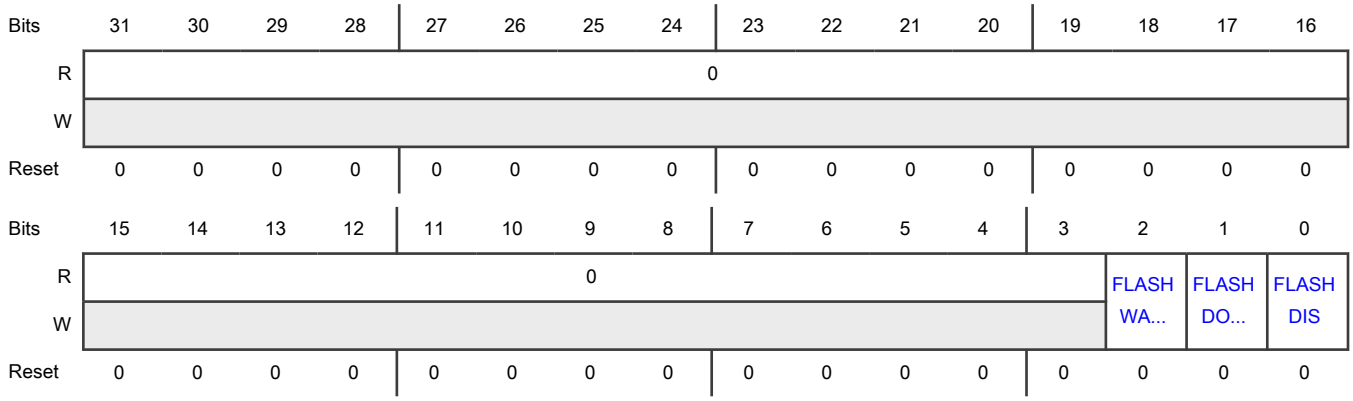
NOTE

This register resets on MAIN warm reset.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram



Fields

Field	Function
31-3 —	Reserved
2 FLASHWAKE	Flash Wake Specifies that when this field becomes 1, an attempt to read the flash memory when it is in Low-Power state because of FLASHCR[FLASHDIS] or FLASHCR[FLASHDOZE], causes the flash memory to exit Low-Power state for the duration of the flash memory access. 0b - No effect 1b - Flash memory is not disabled during flash memory accesses
1 FLASHDOZE	Flash Doze Disables flash memory accesses and places flash memory in Low-Power state whenever the core clock is gated (CKMODE > 0) because of execution of WFI, WFE, or SLEEPONEXIT. Other bus masters that attempt to access the flash memory stalls until the core is no longer sleeping. 0b - No effect 1b - Flash memory is disabled when core is sleeping (CKMODE > 0)
0 FLASHDIS	Flash Disable Places flash memory in Low-Power state when this field becomes 1. Relocate the interrupts out of flash memory before disabling it. 0b - No effect 1b - Flash memory is disabled

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

16.7.1.16 Core Control (CORECTL)

Offset

Register	Offset
CORECTL	110h

Function

Configures options for the core.

NOTE

This register resets on CORE_MAIN warm reset.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															NPIE
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-1 —	Reserved
0 NPIE	Non-maskable Pin Interrupt Enable Enables or disables the pin interrupt. You can write to this field only when NPIE = 0. 0b - Disables 1b - Enables

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

16.7.1.17 Debug Control (DBGCTL)

Offset

Register	Offset
DBGCTL	120h

Function

Configures options for debug.

NOTE

This register resets on MAIN cold reset.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															SOD
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-1 —	Reserved
0 SOD	Sleep Or Debug Configures whether the debug remains enabled when core sleeps. 0b - Remains enabled 1b - Disabled

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Chapter 17
Reset

17.1 Reset sources

The following reset sources exist in this chip:

Table 94. Reset sources

Reset source	Description
Power	System Power-On Reset (POR)
	System Low-Voltage Detect (LVD)
	System High-Voltage Detect (HVD)
External	Wake Up Unit (WUU)
	External pin reset (RESET_b)
Internal	• Watchdog (WDOG) reset • Software (SW) reset • System Clock Generator (SCG-lite) Loss of Clock • Reset Acknowledge timeout (RSTACK) • Low Power Acknowledge (LPACK) • Core Lockup (LOCKUP) • Debug Mailbox

Each system reset source has a corresponding bit that is maintained in the System Reset Status register through reset. This information can be used to take appropriate action when coming out of a reset.

17.2 Power reset sources

To ensure predictable operation of the chip, the supply voltages must remain in their specified ranges during execution. Because this is not possible during power-up and power-down, the chip implements multiple voltage monitors to detect when the supply is outside an acceptable range. When this occurs, the voltage monitor asserts a reset to halt execution and prevent unexpected behavior. The voltage monitors detect power-on events as well as low and high voltage events.

17.2.1 POR reset

The system domain implements a “POR Only Reset (POR)” which only asserts on VDD_SYS analog POR. This reset source is only used for resetting the registers used by the LVD/HVD detectors. The POR voltage monitor is used to ensure that the voltage applied to the system is high enough for other analog modules such as the bias circuits and low voltage monitors to operate correctly. When voltage is initially applied to the chip or when the supply voltage drops below the POR falling threshold, the POR monitor triggers the POR reset condition to the chip.

The POR reset condition asserts POR reset to all power domains. A POR reset sets the appropriate bits in the CMC registers so that software can determine that a POR reset has occurred. See Block diagram.

17.2.2 Low-Voltage Detect (LVD)

The LVD monitors are used to ensure that the voltage supplies to the device are high enough for the logic and memories using those power supplies to operate correctly. When voltage is initially applied to the device, the POR voltage monitor holds the LVD

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

monitors in reset until the voltages are high enough. The LVD monitors continue to hold the device in reset until all monitoring voltages have risen above the respective default LVD Low thresholds (VLVDL). The LVD monitor is enabled by default.

When enabled, the LVD monitor generates a reset when the voltage drops below the V_{LVD} threshold. The voltage threshold at which the LVD reset asserts is programmable. This ensures that the chip halts operation and remains in reset while the monitored voltage is not within the desired range.

The LVD monitors are available in the Active, Sleep, and Deep Sleep modes and are disabled in Deep Power Down modes. See Module operation in low power modes for the available modes.

The individual LVD monitor resets are combined into a single LVD reset trigger to the Core Mode Controller (CMC). The LVD reset condition asserts a reset to all power domains with only the POR/LVD/HVD detection logic remaining operational. A LVD reset will set the appropriate bits in the CMC registers so that software can determine that a LVD reset has occurred.

17.2.3 High-Voltage Detect (HVD)

The HVD monitors are used to ensure that the voltage supplies to the device are within operation range for the logic and memories using those power supplies. The HVD monitor is disabled by default, but when the HVD circuit is enabled, the HVD monitor will generate a reset when the voltage rises above the VHVDL threshold. The voltage threshold at which the HVD reset asserts is programmable. This ensures that the device halts operation and remains in reset while the monitored voltage is not within the desired range.

The HVD monitors are available in the Active, Sleep, and Deep Sleep modes. The HVD monitors are disabled in Deep Power Down modes.

The individual HVD monitor resets are combined into a single HVD reset trigger to the Core Mode Controller (CMC). The HVD reset condition asserts a reset to all power domains with only the POR/LVD/HVD detection logic remaining operational. A HVD reset is combined with LVD reset, and will set the appropriate bits in the CMC registers so that software can determine that a HVD reset has occurred.

17.3 External reset sources

External resets are provided so that the chip can be reset or wake up from very low power states. This allows the chip to start at the correct time from a known state.

17.3.1 External pin reset (RESET_b)

The RESET_b pin is a bi-directional open-drain pin with internal pull-up resistor. The RESET_b pin is functional in two modes:

- During active and low power modes, the RESET_b pin can be asserted externally to force the chip into Pin reset condition.
- During reset, the RESET_b drives low until the chip has completed hardware initialization, at which point the RESET_b pin is released. If the RESET_b pin is asserted externally, then the chip will remain in reset until the RESET_b input is pulled high.

The RESET_b pin implements a digital filter that software can configure to filter out glitches on the RESET_b pin that are less than 1-32 CMC clock cycles. The filter is bypassed in low power modes when the CMC clock is disabled.

The RESET_b is multiplexed on P1_29 pin on this device. RESET_b is the default function of this GPIO pin. Asserting RESET_b wakes the device from any mode.

The PIN reset condition triggers the CMC to assert a Warm reset to all power domains. A Pin reset sets the appropriate fields in the CMC registers so that software can determine that a Pin reset has occurred.

17.3.2 Wake-up (WAKEUP)

The Wake-Up Unit (WUU) provides for a number of external pins and on-chip peripherals to wake the chip from Deep Power Down mode. See the WUU chapter for a list of external and internal reset sources connected to the WUU.

On a wakeup from Deep Power Down mode, the power management logic triggers a WAKEUP reset in the power domains that had been powered off.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

The WAKEUP reset condition triggers the CMC to assert a POR reset to the power domains that were powered down. The system voltage domain is not affected by a WAKEUP reset. A WAKEUP reset sets the appropriate fields in the CMC registers so that software can determine that a WAKEUP reset has occurred.

17.4 Internal reset sources

Internal resets are provided so that the device can be reset when certain erroneous conditions are detected. This allows the device to recover from the erroneous conditions and restart from a known state.

17.4.1 Windowed Watchdog timer (WDOG) reset

The WDOG reset is generated by the WWDT module on the chip.

The WWDT monitors the operation of the system by receiving periodic communication from the software. This communication is generally known as servicing or refreshing the WDOG. If this periodic servicing does not occur, then the Watchdog triggers a WDOG reset condition.

The WDOG reset condition triggers the CMC to assert a WARM reset to all power domains. A WDOG reset sets the appropriate fields in the CMC registers so that software can determine that a WDOG reset has occurred.

17.4.2 Software (SW) reset

During execution, if the software detects erroneous operation, the software can initiate a reset to restart the chip from a known state.

The SW reset condition, which is generated by SYSRESETREQ, triggers the CMC to assert a WARM reset to all power domains. A SW reset sets the appropriate fields in the CMC registers so that software can determine that a SW reset has occurred. For details, refer System Reset Status Register (SRS) in the CMC chapter.

17.4.3 System Clock Generation (SCG)

The SCG module contains a clock monitor, which, if enabled, detects a loss of the external clock. A loss of external clock triggers the SCG reset condition.

The SCG reset condition asserts a WARM reset in all power domains. The CMC_SRS[SCG] and CMC_SRS[WARM] status bits are set on a SCG reset condition.

For details on enabling the clock monitor, see the SCG chapter. To prevent unexpected reset events, clock monitors must be disabled before entering any low-power modes that the external clocks are not present.

17.4.4 Reset acknowledge timeout (RSTACK)

The reset state machine includes a timeout counter that is triggered by the reset state machine not progressing within 65536 cycles of the clk_1m clock, which triggers the RSTACK reset condition.

The RSTACK reset condition triggers the CMC to assert a WARM reset to all power domains. An RSTACK reset sets the appropriate fields in the CMC registers so that software can determine that an RSTACK reset has occurred.

17.4.5 Low power acknowledge timeout (LPACK)

The low power entry state machine includes a timeout counter that is triggered if a module does not acknowledge entry into a low power mode after 65536 cycles of the clk_1m clock; this triggers the LPACK reset condition.

The LPACK reset condition triggers the CMC to assert a WARM reset to all power domains. An LPACK reset sets the appropriate fields in the CMC registers so that software can determine that an LPACK reset has occurred.

17.4.6 Lock up (LOCKUP)

The Cortex-M23 core enters the lockup state as a result of certain illegal operations, which triggers the LOCKUP reset condition.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

The LOCKUP reset condition triggers the CMC to assert a WARM reset to all power domains. A LOCKUP reset sets the appropriate fields in the CMC registers so that software can determine that a LOCKUP reset has occurred.

17.4.7 Debug Mailbox (DBGMB)

DBGMB (ISP-AP) can initiate a reset request from a connected debugger. This triggers the DBGMB reset condition.

The DBGMB reset condition triggers the CMC to assert a WARM reset to all power domains. A DBGMB reset sets the appropriate fields in the CMC registers so that software can determine that a DBGMB reset has occurred.

17.5 Reset type

This section describes the different resets that can be generated on this chip and their respective reset sequences. Reset conditions on this chip are categorized into two types, Cold and Warm resets.

17.5.1 Cold reset

A cold reset is a reset that only asserts when a power domain is powered up (Power On Reset, Deep Power Down wakeup, etc) or one of the supply voltages is not valid (LVD, HVD, supply glitch detect). For simplicity, all LVD/HVD assert the Cold Reset for all power domains. Each power domain has its own "Cold Reset" which is used by any logic in that domain that needs a reset that does not assert on a warm reset source. The system domain also implements a "POR Only Reset" which only asserts on VDD_SYS analog POR. This reset source is only used for resetting the registers used by the LVD/HVD detectors. A POR does not guarantee the contents of on-chip SRAM, except for a Deep Power Down wakeup and then only for the SRAM that is configured to retain state.

17.5.1.1 POR reset sequence

The following steps are performed as part of the POR (or LVD/HVD) sequence:

1. RESET_b pin is driven low and internal reset signals assert.
2. POR and LVD signals negate and clocks are enabled in their default configuration.
3. Internal reset remains asserted for 32 clock cycles.
4. Internal reset to Flash and Fuse controller negate and their initialization sequences commence.
5. Initialization sequences for Flash and Fuse complete.
6. Internal trim registers are loaded and RESET_b pin is tri-stated.
7. Reset state machine waits for RESET_b pin input to pull high or drive high externally.
8. Internal reset signals negate.
9. Core exits reset and fetches the initial program counter and stack pointer from Boot ROM.

When the chip wakes up from Deep Power Down, the reset sequence is slightly different from the POR reset sequence. They are described in the subsequent sections.

17.5.1.2 Deep Power Down (DPD) reset sequence

The following steps are performed as part of the Deep Power Down wakeup sequence:

1. Internal reset signals assert.
2. Wakeup signal negates and clocks are enabled in their default configuration.
3. Internal reset remains asserted for 32 clock cycles.
4. Internal reset to Flash and Fuse controller negate and their initialization sequences commence.
5. Initialization sequences for Flash and Fuse complete.
6. Internal trim registers are loaded.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

7. Internal reset signals negate.
8. Core exits reset and fetches the initial program counter and stack pointer from Boot ROM.

The RESET_b pin is not driven low by a Deep Power Down reset sequence unless the RESET_b pin was used as the wake up source.

A wakeup from Deep Power Down via the RESET_b pin will follow the warm reset sequence.

17.5.2 Warm reset

A warm reset is a reset that asserts on either:

- Cold Reset for that power down
- Any of the warm reset sources

Any of the warm reset sources will trigger a warm reset for every power domain. An “early” warm reset is generated for some of the power domain, this reset asserts at the same time as the normal reset but negates early to allow processes that need to operate during reset to complete. This includes the MSF loading of IFR trim data.

A WARM reset re-initializes the chip to a known state once it has already been running for a period of time.

A WARM reset resets most of the logic in each power domain. Warm resets are divided into fatal reset sources and non-fatal reset sources. See the following table to know which CMC reset sources are fatal/non-fatal. Also see CMC chapter for details.

CMC reset source	Fatal Reset
Reset pin	N
Debug AP (debug mailbox)	N
Reset Monitor	Y
Low Power Monitor	N
Loss of Clock / Lock	N
WDOG0/1 reset	N
CM23 reset request	N
CM23 lockup	N

Non-fatal reset sources can be configured to generate an interrupt instead of the WARM reset. If software can clear the non-fatal reset source (including status flag) within 64 ms then the WARM reset is averted. Non-fatal resets retain the contents of on-chip SRAM.

Fatal reset sources cannot be configured to generate an interrupt, and do not guarantee the contents of on-chip SRAM.

The following steps are performed as part of the warm reset sequence:

1. RESET_b pin drives low and internal reset signals assert.
2. Clocks are enabled in their default configuration.
3. Internal reset remains asserted for 32 clock cycles.
4. Internal reset to Flash and Fuse controller negate and their initialization sequences commence.
5. Initialization sequences for Flash and Fuse complete.
6. Internal trim registers are loaded and RESET_b pin is tri-stated.
7. Reset state machine waits for RESET_b pin input to pull high or drive high externally.
8. Internal reset signals negate.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Core exits reset and fetches the initial program counter and stack pointer from Boot ROM.

17.5.3 Peripheral SW reset

Peripherals can be reset by SW. The chip implements Peripheral reset control registers in the System Controller (SYSCON) chapter.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Chapter 18
Lifecycle States

18.1 Overview

This device implements a security lifecycle state model that governs its operational behavior. The current lifecycle state determines the device functionality, debug port availability, and access to security-sensitive assets. The lifecycle state is controlled by the LIFECYCLE_STATE value and LIFECYCLE_STATE_DP value that stored in CMPA region within user IFR (IFR0). LIFECYCLE_STATE is located at offset 0x1000 within the CMPA region, and LIFECYCLE_STATE_DP is located at offset 0x1004. Make sure that both values are exactly the same.

The lifecycle state can be advanced by correctly programming the values of LIFECYCE_STATE and LIFECYCLE_STATE_DP. The device can return to Develop lifecycle using a password-protected mass erase command in In-field1 or In-field2 lifecycle state.

The bootloader checks the lifecycle state during initialization and adjusts its behavior accordingly. Based on the lifecycle state, the bootloader controls access of debug port and manages the availability of ISP commands. If the device is at Bricked state ,the bootloader locks the device, preventing further operations.

18.2 Lifecycle states

Table below lists the lifecycle states and shows the debug port access for each state.

Table 95. Lifecycle states

State	Type	Lifecycle	Lifecycle_Status	Debug port enabled?	ISP commands	Description
Develop	Customer	0x3	0xFFFF_FFFF/ 0x9635_FC03	Yes, after a successful boot	All	Initial customer development state after leaving NXP manufacturing
In-field1	Customer	0x7	0x9635_F807	Debug authentication required	Limited	In-field application state for end-customer use
In-field2	Customer	0xF	0x9635_F00F	No	Limited	Optional In-field application state for end-customer use
In-field Locked	Customer	0xCF	0x9635_30CF	No	Limited	Alternative in-field application state that prevents use of field return/failure analysis states. The rest of the behavior of the device is the same as the In-field2 state
Bricked	Customer	0xFF	0x9635_A55A	No	Disabled	Bricked state to prevent device use.

18.3 Lifecycle state transitions

The diagram below shows the lifecycle states and transitions, with the device state shown provisioned and/or configured as it is expected to be in the respective lifecycle state.

All information provided in this document is subject to legal disclaimers.
© 2026 NXP B.V. All rights reserved.

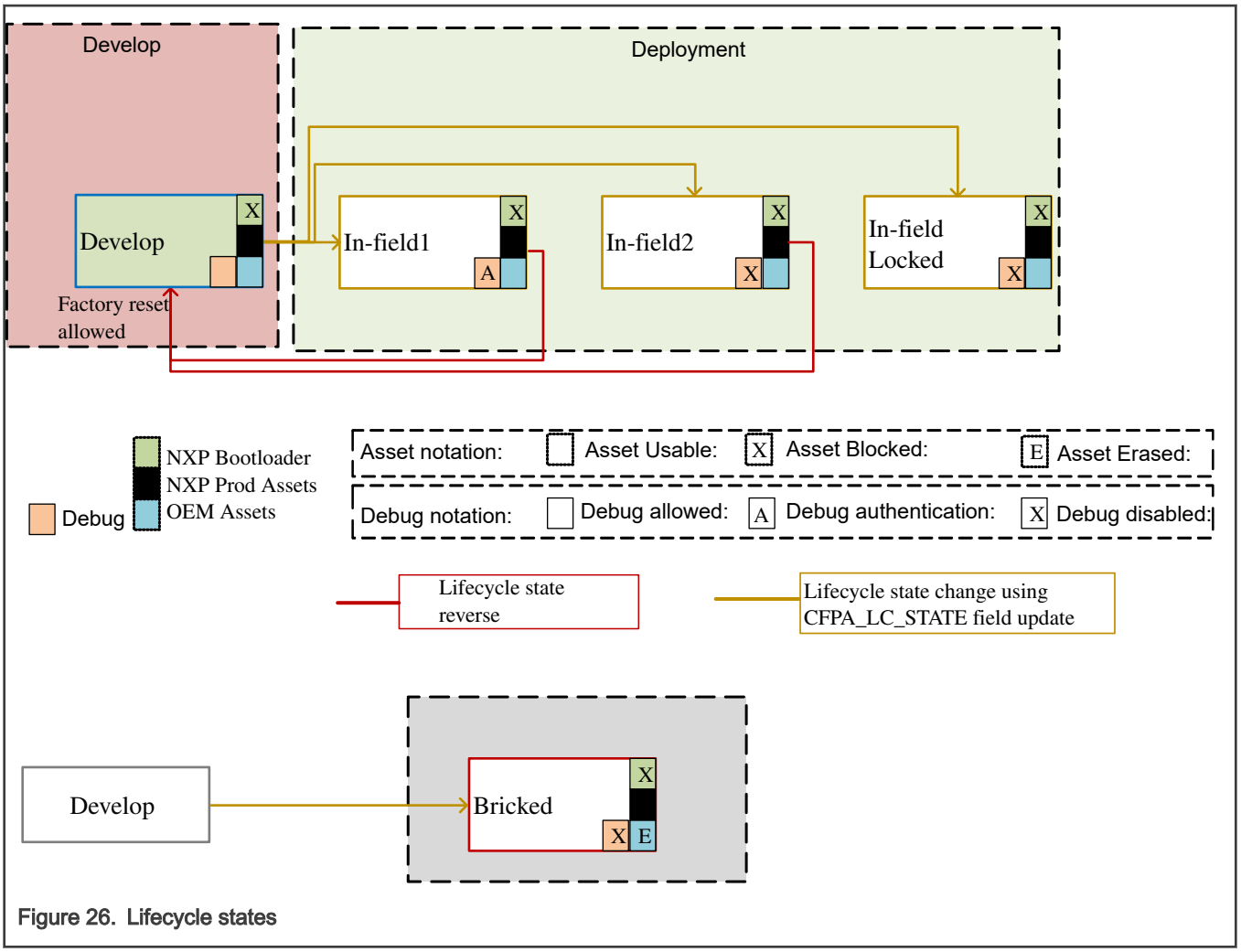


Figure 26. Lifecycle states

18.4 Customer lifecycle states

The following sections give descriptions of the customer lifecycle states:

- Develop
- In-field1
- In-field2
- In-field Locked
- Bricked

The customer lifecycle states are described in detail along with information on all the state transitions.

18.4.1 Develop lifecycle state

The devices delivered from NXP are in the Develop lifecycle state. In this state, all customer chip functionality is accessible. This mode is recommended for early software development.

In the Develop state:

- Debug port is open after boot application by default.

All information provided in this document is subject to legal disclaimers.
© 2026 NXP B.V. All rights reserved.

- Bootloader performs authentication based on image type of boot image and boots if the authentication passes. If the authentication fails, the bootloader locks the device and prevents further operations if the IMG_CHECK_ENF field in CMPA is enabled in the current lifecycle. Otherwise, bootloader enters ISP mode.
- All ISP commands are available in this lifecycle state (GetProperty, SetProperty, WriteMemory, ReadMemory, FlashEraseRegion, Execute, FlashEraseAll, FillMemory and Reset).

From the Develop lifecycle, the device can be advanced to In-field1, In-field2, In-field Locked, or Bricked states.

18.4.2 In-field lifecycle states

The in-field state is the primary state to be used for deployment of products to end-customers in the field.

In the in-field states:

- Debug port is closed by default. The debug authentication process can be used to reopen debug port in In-field1 state (LIFECYCLE_STATE= 0x9635_F807). In other in-field states, the debug port cannot be reopened.
- In the in-field lifecycle states, image authentication behavior is the same as Develop state.
- Limited ISP commands are allowed. In the In-field1 or In-field2 state (LIFECYCLE_STATE = 0x9635_F807/0x9635_F00F), the available ISP commands includes GetProperty, Reset, SetProperty, FlashEraseAll and FillMemory. In the In-field Locked state (LIFECYCLE_STATE = 0x9635_30CF), the available ISP commands are GetProperty and Reset.

From the In-field1 or In-field2 lifecycle state, the device can transition back to the Develop lifecycle state. In-field Locked state device cannot return to the previous state.

18.4.3 Lifecycle states transition from Develop to in-field

Devices can be moved from Develop to one of the in-field lifecycle states when development is complete and product is ready to be deployed in the field.

Before transitioning from Develop to one of the in-field lifecycle states, all configurations in the CMPA region should be programmed correctly.

In the Develop lifecycle state, LIFECYCLE_STATE field and LIFECYCLE_STATE_DP field can be programmed using the ISP command (WriteMemory) to advance the lifecycle state. When the correct in-field state is successfully programmed into both LIFECYCLE_STATE field and LIFECYCLE_STATE_DP field, the device will transition to the target in-field lifecycle state after a warm reset.

NOTE

The Password Area header and the correct CRC value must be filled in on Develop lifecycle; otherwise, the lifecycle cannot be rolled back from in-field to Develop.

18.4.4 Invalid lifecycle state

If the value in the LIFECYCLE_STATE field does not match the value in the LIFECYCLE_STATE_DP field, or if either value is invalid, the bootloader treats the device as being in Invalid lifecycle state and goes to ISP mode.

In the Invalid state:

- Debug port is closed.
- Limited ISP commands are allowed (GetProperty, Reset, SetProperty, FlashEraseAll and FillMemory).
- The invalid state value (0xBADBADBA) can be retrieved using the ISP command (GetProperty <kPropertyTag_SecurityState>).

18.4.5 Failure Analysis (FA) lifecycle state

The FA state is used for products that are returned to NXP for testing and failure analysis. In this device, the Develop state is used as the FA state. Devices can be moved from In-Field1, In-Field2 state or Invalid state to the FA lifecycle state using ISP commands (FillMemory and FlashEraseAll).

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

To transition from In-Field1, In-Field2 state or Invalid state to FA state:

1. Call FillMemory command to fill four consecutive 32-bit passwords into ERASE_PASSWORD_ADDRESS [0x400E86B8,0x400E86C7].
2. Call FlashEraseAll command. Bootloader will compare the passwords at ERASE_PASSWORD_ADDRESS with the value of the ERASE_PASSWORD field in the CMPA region. If the comparison passes, the Flash regions [0x0, 0xFFFF], [0x1C000,0x1FFFF], and the IFR0 region [0x01002000, 0x01007FFF] will be erased.
3. Reset.

18.4.6 End-of-life lifecycle state

The Bricked state is the end-of-life state used when a device is permanently removed from service. In this state, the bootloader immediately locks the device upon any reset, and the debug port is disabled.

The Bricked state is the final lifecycle state; therefore, the device cannot be transitioned to any other state once it is in Bricked. Once the device's lifecycle state transitions to the Bricked state, the device becomes permanently disabled.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Chapter 19
Bootloader

19.1 Overview

The bootloader is the initial code stored in the on-chip flash that executes immediately after the device is released from reset. This chapter describes the features supported by the bootloader and its execution flow.

The bootloader provides the capabilities of booting from internal flash and in-system programming (ISP) via a serial interface LPUART, LPI2C or LPSPI. Bootloader also supports Debug Mailbox (DM), enabling user to start debug session when no valid boot image is present.

The bootloader resides in internal flash, occupying a portion of Program Flash and one sector of User IFR (IFR0). The memory map used during bootloader execution is shown in the diagram below.

During bootloader operation, including boot after warm reset, Power-On Reset (POR), and wakeup from deep power-down, the first 5 KB of SRAM (0x2000_0000–0x2000_13FF) must be reserved and must not be used by other components. After a successful boot, the reserved SRAM region is released and becomes available for general use. The SRAM is ECC-protected and must be properly initialized before use. If RAM ECC initialization is not enabled in the CMPA settings, the bootloader does not perform full SRAM initialization during the boot process.

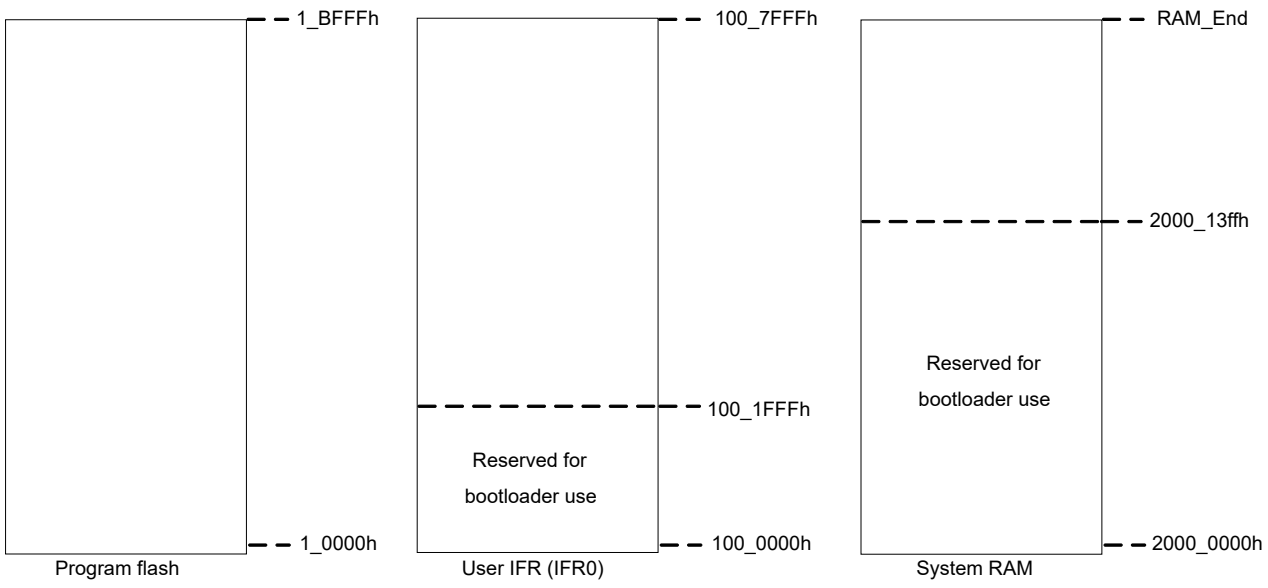


Figure 27. Bootloader memory usage

Bootloader configurations are stored in the Customer Manufacturing/Factory Programming Area (CMPA) within IFR0 and the bootloader behavior is determined by these settings.

19.2 Boot flow

The device always starts the bootloader after a reset. Key characteristics of the boot flow include:

- Default operating frequency: 48 MHz
- The bootloader determines whether to enter ISP mode based on the scratch register (runBootloader API) and ISP pin (P3_29).

All information provided in this document is subject to legal disclaimers.
© 2026 NXP B.V. All rights reserved.

- Debug Mailbox (DM) can be selected after the debugger requests resynchronization and resets the chip

By default, the bootloader attempts a normal boot. If this fails, it goes to ISP mode. For normal boot, the bootloader checks for the presence of a boot image and performs an image validity check. For power-on reset and system reset, the bootloader does basic checks, CRC32 validation or MISR authentication for boot image depending on CMPA configuration. For waking up from Deep Power Down, the bootloader performs a CRC32 check on a portion of the image. If the boot image is valid, the bootloader jumps to the image entry point.

The figure below illustrates the high-level boot flow.

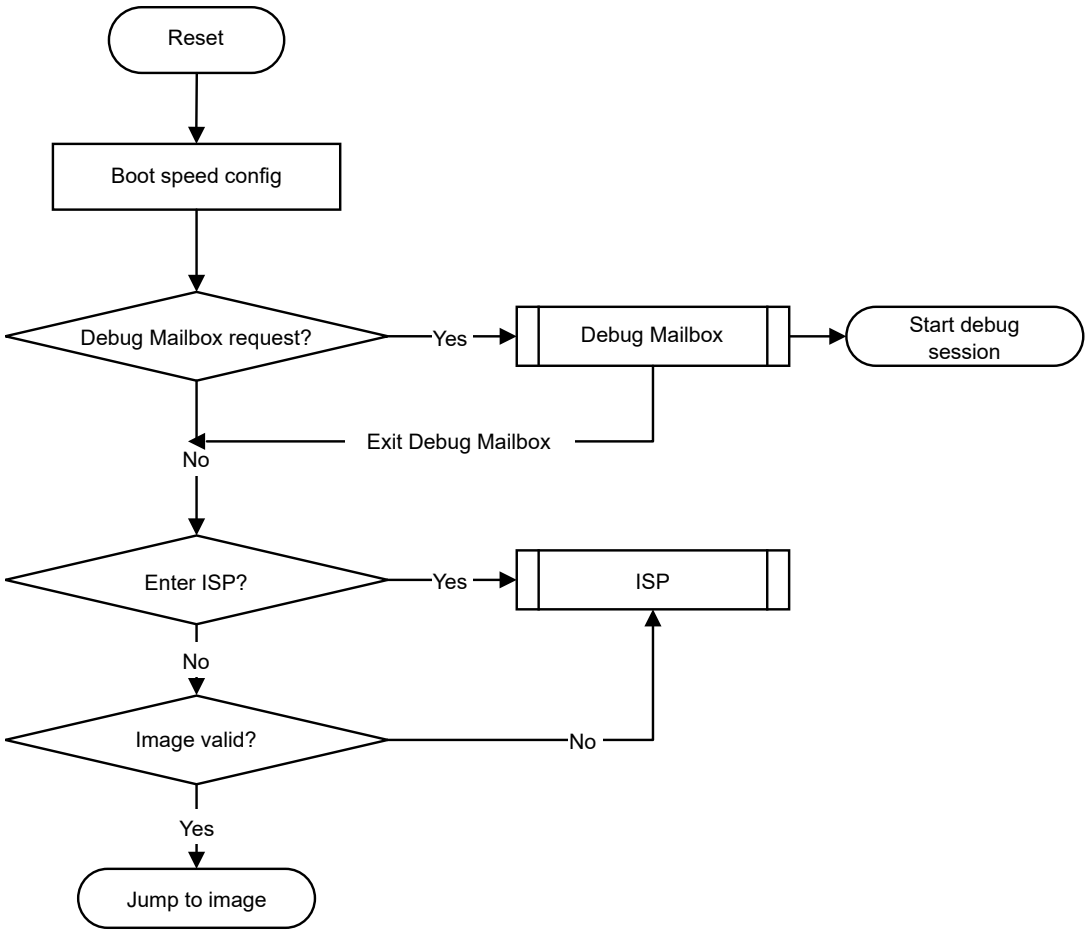


Figure 28. Boot flow

19.2.1 Boot image

The bootloader supports the following image types when the IMG_CHECK_ENF (Image Check Enforcement) bits allow all image types. By default, IMG_CHECK_ENF is configured to permit all image types:

- Plain image
- CRC image
- MISR image

Verification of the CRC32 checksum or MISR signature is performed only if the image specifies a valid image type at offset 24h in the vector table. The table below provides the details of the image type word.

If the IMG_CHECK_ENF bits restrict the boot image type to CRC image or MISR image, then only the corresponding verification is executed. The boot process will succeed only if the verification passes.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

If the IMG_CHECK_ENF bit restricts the boot image type to a MISR image, but the PASSWORD_HEADER does not match, then IMG_MISR_SEED uses the default 0x0.

Table 96. Image Type (word at offset 24h)

Bit	Field	Value
31:8	Reserved	Set to 0
7:0	Image Type	0h: plain image 5h: CRC image Bh: MISR image

19.2.1.1 Plain image

Bootloader supports plain image when IMG_CHECK_ENF (Image Check Enforcement) allows all image types. A plain image must meet the following requirements:

- The first word contains the initial Stack Pointer (SP) value
- The second word contains Program Counter (PC) value

The image boots successfully if these two values are valid.

19.2.1.2 CRC image

Figure below illustrates the structure of CRC image:

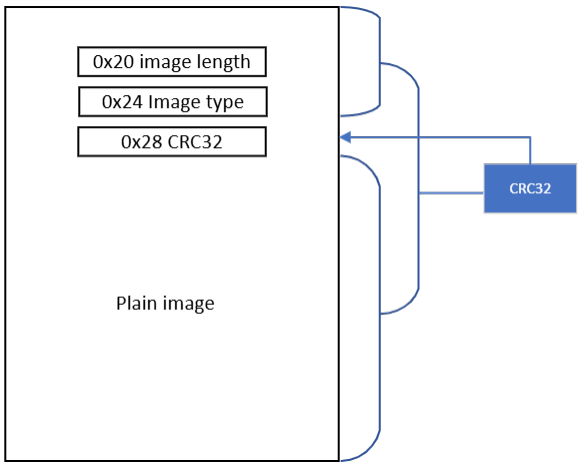


Figure 29. Structure of CRC images

The image length field at offset 20h specifies the total length of the image in bytes. This value is used during CRC verification of the boot image. The CRC calculation process starts from the beginning of the image (offset 0h) and continues up to the number of bytes indicated by the image length field, excluding the CRC32 value word located at offset 28h. The computed CRC32 result is then compared with the CRC32 value stored in the image. If the values match, the image is considered valid; otherwise, the image is invalid.

19.2.1.3 MISR image

Figure below illustrates the structure of MISR image:

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

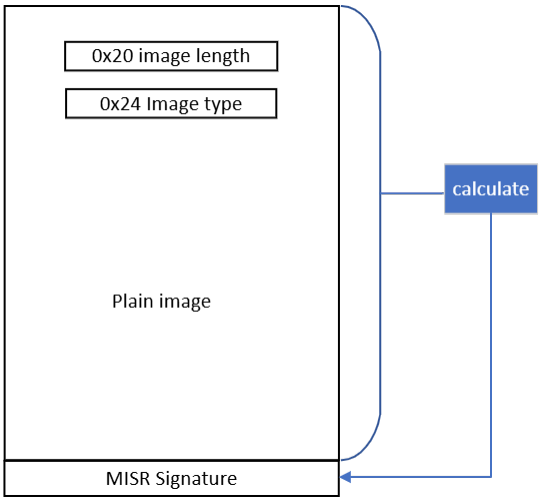


Figure 30. Structure of MISR images

The MISR signature must be placed at a 128-byte aligned address, and all image data preceding the signature must be padded to maintain 128-byte alignment. The image length field at offset 20h specifies the total image size in bytes, including the MISR signature. The length, minus 16 bytes, must be aligned to a 128-byte boundary.

During boot, the bootloader performs MISR verification using the MISR seed stored in CMPA. The image is considered valid only if the MISR verification succeeds.

19.2.2 CMPA configuration options

The Customer Manufacturing Area (CMPA) is a dedicated region within User IFR (IFR0) used to configure the bootloader features. CMPA is located in the sector 1 of IFR0 (Base address: 0100_2000h) and consists of three parts:

- Bootloader configuration area
- Password area
- Lifecycle area

The Bootloader Configuration Area and Password Area can be freely programmed and read while the device is in the Develop lifecycle. The bootloader considers the configuration valid only when both the CMPA header and the PWD header are correctly programmed. Once the device transitions to any In-field lifecycle state, these regions can no longer be updated.

The Lifecycle Area can be programmed using either the ISP write-memory command or the DBGMB Advance Lifecycle command while in the Develop lifecycle. Before executing a lifecycle programming command, valid headers and CRC values must be present in both the Bootloader Configuration Area and the Password Area.

NOTE

The CMPA region can only be erased using the flash-erase-all command. This operation is not permitted when the device is in an In-field Locked lifecycle state.

The tables below detail the fields for each CMPA area.

Table 97. CMPA bootloader configuration area

CMPA configuration field	Offset	Bit	Description
Header	0h	[31:16]	CMPA user configuration area header should be set to 5963h.

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table 97. CMPA bootloader configuration area (continued)

CMPA configuration field	Offset	Bit	Description
			The CMPA user configuration data is considered valid only when the CMPA user configuration header maker is properly configured and valid.
BOOT_CFG[ISP_FT_ENTRY]	4h	[15:14]	Disable ISP mode entry on image authentication failure. 01 - ISP entry disabled. 00,10, 11 - ISP entry allowed.
BOOT_CFG[ISP_API_ENTRY]	4h	[13:12]	Disable ISP mode entry through ROM API call. ISP mode can be entered through ROM API invocation 01 - ISP entry disabled. 00,10, 11 - ISP entry allowed.
BOOT_CFG[ISP_PIN_ENTRY]	4h	[9:8]	Disable ISP mode entry through pin assertion. 01 - ISP entry disabled. 00,10, 11 - ISP entry allowed.
BOOT_CFG[ISP_SPI_EN]	4h	[2]	Enable SPI interface in ISP boot. 1 - Enable 0 - Disable
BOOT_CFG[ISP_I2C_EN]	4h	[1]	Enable I2C interface in ISP boot. 1 - Enable 0 - Disable
BOOT_CFG[ISP_UART_EN]	4h	[0]	Enable UART interface in ISP boot. 1 - Enable 0 - Disable
BOOT_TIMERS[POWERDOWN_TIMEOUT_SECS]	Ch	[15:0]	Power down timeout:Timeout value in seconds. When a non-zero value is programmed in this field ROM uses it as idle timeout value to enter power-down state to conserve power
ISP_UART_CFG[UART_TX_FUNC_SLOT]	10h	[27:24]	Identifies the pin mux function slot.
ISP_UART_CFG[UART_TX_PIN]	10h	[23:16]	Override default UART TX ISP pin. Identifies the pin to be used as UART_TX pin. [4:0] GPIO Pin number [7:5] GPIO port number
ISP_UART_CFG[UART_ISP_INSTENCE_SEL]	10h	[15:12]	Identifies the LPUART instance used for UART ISP mode.
ISP_UART_CFG[UART_RX_FUNC_SLOT]	10h	[11:8]	Identifies the pin mux function slot.

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table 97. CMPA bootloader configuration area (continued)

CMPA configuration field	Offset	Bit	Description
ISP_UART_CFG[UART_RX_PIN]	10h	[7:0]	Override default UART RX ISP pin. Identifies the pin to be used as UART_RX pin. [4:0] GPIO Pin number [7:5] GPIO port number
ISP_I2C_CFG[I2C_SDA_FUNC_SLOT]	14h	[27:24]	Identifies the pin mux function slot.
ISP_I2C_CFG[I2C_SDA_PIN]	14h	[23:16]	Override default I2C SDA ISP pin. Identifies the pin to be used as I2C_SDA pin. [4:0] GPIO Pin number [7:5] GPIO port number
ISP_I2C_CFG[I2C_ISP_INSTANCE_SEL]	14h	[15:12]	Identifies the LPI2C instance used for LPI2C ISP mode.
ISP_I2C_CFG[I2C_SCL_FUNC_SLOT]	14h	[11:8]	Identifies the pin mux function slot.
ISP_I2C_CFG[I2C_SCL_PIN]	14h	[7:0]	Override default I2C SCL ISP pin. Identifies the pin to be used as I2C_SCL pin.[[4:0] GPIO Pin number [7:5] GPIO port number
ISP_SPI_CFG0[SPI_MOSI_FUNC_SLOT]	18h	[27:24]	Identifies the pin mux function
ISP_SPI_CFG0[SPI_MOSI_PIN]	18h	[23:16]	Override default SPI_MOSI ISP pin. Identifies the pin to be used as SPI_MOSI pin. [4:0] GPIO Pin number [7:5] GPIO port number
ISP_SPI_CFG0[SPI_ISP_INSTANCE_SEL]	18h	[15:12]	Identifies the LPSPI instance used for LPSPI ISP mode.
ISP_SPI_CFG0[SPI_SCK_FUNC_SLOT]	18h	[11:8]	Identifies the pin mux function slot.
ISP_SPI_CFG0[SPI_SCK_PIN]	18h	[7:0]	Override default SPI_SCK ISP pin. Identifies the pin to be used as SPI_SCK pin. [4:0] GPIO Pin number [7:5] GPIO port number
ISP_SPI_CFG1[ISP_SPI_SSEL_X]	1Ch	[29:28]	SPI chip select number. LPSPI interfaces supports up to four chip selects.
ISP_SPI_CFG1[SPI_SSEL_FUNC_SLOT]	1Ch	[27:24]	Identifies the pin mux function slot.

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table 97. CMPA bootloader configuration area (continued)

CMPA configuration field	Offset	Bit	Description
ISP_SPI_CFG1[SPI_SSEL_PIN]	1Ch	[23:16]	Override default SPI_SSEL ISP pin. Identifies the pin to be used as SPI_SSEL pin. [4:0] GPIO Pin number [7:5] GPIO port number
ISP_SPI_CFG1[SPI_MISO_FUNC_SLOT]	1Ch	[11:8]	Identifies the pin mux function slot.
ISP_SPI_CFG1[SPI_MISO_PIN]	1Ch	[7:0]	Override default SPI_MISO ISP pin.Identifies the pin to be used as SPI_MISO pin. [4:0] GPIO Pin number [7:5] GPIO port number
ISP_MISC_CFG[ISP_UART_CUST]	20h	[31:30]	Use customer defined UART ISP pins. 01: Customer defined. 00, 10, 11: Default ROM defined pins.
ISP_MISC_CFG[ISP_I2C_CUST]	20h	[29:28]	Use customer defined I2C ISP pins. 01: Customer defined. 00, 10, 11: Default ROM defined pins.
ISP_MISC_CFG[ISP_SPI_CUST]	20h	[27:26]	Use customer defined SPI ISP pins. 01: Customer defined. 00, 10, 11: Default ROM defined pins.
ISP_MISC_CFG[QUICK_GPIO_CUST]	20h	[25:24]	Use customer defined quick gpio configuration. 01: Customer defined by the following QUICK_SETGPIO and QUICK_CLR_GPIO. 00, 10, 11: Default gpio setting.
ISP_MISC_CFG[I2C_SLAVE_ADDR]	20h	[7:0]	I2C slave address A 7-bit Address used for selecting our device on shared I2C bus system. By default, Bootloader uses 10h as slave address. If this address conflicts with another slave on board customer could use this field to override the address.
BOOT_IMG_CFG[RAM_ECC_CLEAR]	24h	[31:30]	Clear ECC for the whole RAM at boot stage. 00, 10, 11 - Not clear 01 - clear
BOOT_IMG_CFG[TRNG_INIT]	24h	[29:28]	Enable the TRNG for customer at boot stage. 00, 11 - Not clear 01 - Init TRNG with Low quality entropy

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table 97. CMPA bootloader configuration area (continued)

CMPA configuration field	Offset	Bit	Description
			10 - Init TRNG with High quality entropy
BOOT_IMG_CFG[ACTIVE_IMG_PROT]	24h	[15:14]	Protection of active image. This field defines protection of flash area occupied by the active image. Only applicable to internal flash. 10 - Protection is defined using the CMPA FLASH_ACL settings and IFR0_ACL_settings. Other - GLBAC0 (RWX) is used. CMPA FLASH_ACL settings and IFR0_ACL_settings are ignored.
BOOT_IMG_CFG[FAST_WAKEUP_EN]	24h	[9:8]	This field defines whether to go fast wakeup flow when wake up from Deep Power Down mode. 01 - Bootloader go low power wakeup path when wakeup from DPD mode.Note that SRAM should be retained when set this field. other - Bootloader go normal boot flow when wakeup from DPD mode.
BOOT_IMG_CFG[LP_CRC_EN]	24h	[7:6]	This field defines whether do CRC32 check for low power wakeup path. 01 - CRC32 check is needed. other - CRC32 check is disable.
BOOT_IMG_CFG[IMG_CHECK_ENF]	24h	[1:0]	Image check enforcement. This field defines the minimum image verification procedure (CRC32, MISR). 00 - All Image types are allowed. 01 - CRC image is enforced. 10 - MISR image is enforced. 11 - All Image types are allowed.
LP_CTX_ADDR_CFG[LOW_POWER_CONTEXT_ADDRESS]	28h	[31:0]	Low-power content address.
QUICK_SET_GPIO_0	30h	[31:0]	Drive GPIO 0 port pins high after reset. Each bit corresponds to the pin in GPIO port 0. When set ROM drives the corresponding pin high as soon as possible. By default, most pins come-up as tri-stated inputs. This feature allows customer to specify active drive pins soon after reset instead of waiting till complete boot. Drive GPIO 0 port pins low.
QUICK_CLR_GPIO_0	34h	[31:0]	Drive GPIO 0 port pins low. See description of QUICK_SET_GPIO_0 field.

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table 97. CMPA bootloader configuration area (continued)

CMPA configuration field	Offset	Bit	Description
QUICK_SET_GPIO_1	38h	[31:0]	Drive GPIO 1 port pins high after reset. Each bit corresponds to the pin in GPIO port 1. When set ROM drives the corresponding pin high as soon as possible. By default, most pins come-up as tri-stated inputs. This feature allows customer to specify active drive pins soon after reset instead of waiting till complete boot. Drive GPIO 1 port pins low.
QUICK_CLR_GPIO_1	3Ch	[31:0]	Drive GPIO 1 port pins low. See description of QUICK_SET_GPIO_1 field.
QUICK_SET_GPIO_2	40h	[31:0]	Drive GPIO 2 port pins high after reset. Each bit corresponds to the pin in GPIO port 2. When set ROM drives the corresponding pin high as soon as possible. By default, most pins come-up as tri-stated inputs. This feature allows customer to specify active drive pins soon after reset instead of waiting till complete boot. Drive GPIO 2 port pins low.
QUICK_CLR_GPIO_2	44h	[31:0]	Drive GPIO 2 port pins low. See description of QUICK_SET_GPIO_2 field.
QUICK_SET_GPIO_3	48h	[31:0]	Drive GPIO 3 port pins high after reset. Each bit corresponds to the pin in GPIO port 3. When set ROM drives the corresponding pin high as soon as possible. By default, most pins come-up as tri-stated inputs. This feature allows customer to specify active drive pins soon after reset instead of waiting till complete boot. Drive GPIO 3 port pins low.
QUICK_CLR_GPIO_3	4Ch	[31:0]	Drive GPIO 3 port pins low. See description of QUICK_SET_GPIO_3 field.
QUICK_SET_GPIO_4	50h	[31:0]	Drive GPIO 4 port pins high after reset. Each bit corresponds to the pin in GPIO port 4. When set ROM drives the corresponding pin high as soon as possible. By default, most pins come-up as tri-stated inputs. This feature allows customer to specify active drive pins soon after reset instead of waiting till complete boot. Drive GPIO 4 port pins low.
QUICK_CLR_GPIO_4	54h	[31:0]	Drive GPIO 4 port pins low. See description of QUICK_SET_GPIO_4 field.
FLASH_ACL_0_7[ACL_SEC_7]	60h	[30:28]	Flash MBC setting: E000h-FFFFh

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table 97. CMPA bootloader configuration area (continued)

CMPA configuration field	Offset	Bit	Description
FLASH_ACL_0_7[ACL_SEC_6]	60h	[26:24]	Flash MBC setting: C000h-DFFFh
FLASH_ACL_0_7[ACL_SEC_5]	60h	[22:20]	Flash MBC setting: A000h-BFFFh
FLASH_ACL_0_7[ACL_SEC_4]	60h	[18:16]	Flash MBC setting: 8000h-9FFFh
FLASH_ACL_0_7[ACL_SEC_3]	60h	[14:12]	Flash MBC setting: 6000h-7FFFh
FLASH_ACL_0_7[ACL_SEC_2]	60h	[10:8]	Flash MBC setting: 4000h-5FFFh
FLASH_ACL_0_7[ACL_SEC_1]	60h	[6:4]	Flash MBC setting: 2000h-3FFFh
FLASH_ACL_0_7[ACL_SEC_0]	60h	[2:0]	Flash MBC setting: 0h-1FFFh
FLASH_ACL_8_9[ACL_SEC_9]	64h	[6:4]	Flash MBC setting: 1E000h-1FFFFh
FLASH_ACL_8_9[ACL_SEC_8]	64h	[2:0]	Flash MBC setting: 1C000h-1DFFFh
IFR0_ACL_8-15[IFR0_ACL_SEC_15]	68h	[30:28]	IFR0 MBC setting: 100_7800h-100_7FFFh
IFR0_ACL_8-15[IFR0_ACL_SEC_14]	68h	[26:24]	IFR0 MBC setting: 100_7000h-100_77FFh
IFR0_ACL_8-15[IFR0_ACL_SEC_13]	68h	[22:20]	IFR0 MBC setting: 100_6800h-100_6FFFh
IFR0_ACL_8-15[IFR0_ACL_SEC_12]	68h	[18:16]	IFR0 MBC setting: 100_6000h-100_67FFh
IFR0_ACL_8-15[IFR0_ACL_SEC_11]	68h	[14:12]	IFR0 MBC setting: 100_5800h-100_7FFFh
IFR0_ACL_8-15[IFR0_ACL_SEC_10]	68h	[10:8]	IFR0 MBC setting: 100_5000h-100_57FFh
IFR0_ACL_8-15[IFR0_ACL_SEC_9]	68h	[6:4]	IFR0 MBC setting: 100_4800h-100_4FFFh
IFR0_ACL_8-15[IFR0_ACL_SEC_8]	68h	[2:0]	IFR0 MBC setting: 100_4000h-100_47FFh
IFR0_ACL_7[IFR0_ACL_SEC_7]	6Ch	[2:0]	IFR0 MBC setting: 100_3800h-100_3FFFh
CMPA_CRC32	70h	[31:0]	For address range (0h - 6Fh)

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table 98. CMPA password area

CMPA password field	Offset	Bit	Description
PASSWORD_HEADER	800h	[31:0]	PASSWORD region header marker should be set to 5057h. The CMPA password data is considered valid only when the CMPA password header marker is properly configured and valid.
DBG_AUTH_PASSWORD	810h	[31:0]	Passwords defined by the user is stored in CMPA and used for debug authentication. All 0b or all 1b password means debug authentication is disabled.
	814h	[62:32]	
	818h	[95:64]	
	81Ch	[127:96]	
IMG_MISR_SEED	820h	[31:0]	A set of seeds defined by the user, which the bootloader can use to verify the integrity of the MISR image.
	824h	[62:32]	
	828h	[95:64]	
	82ch	[127:96]	
ERASE_PASSWORD	830h	[31:0]	Passwords defined by the user to use MASS ERASE command in ISP at in-field1 or in-field2 lifecycle.
	834h	[62:32]	
	838h	[95:64]	
	83Ch	[127:96]	
PASSWORD_AREA_CRC32	840h	[31:0]	PASSWORD_AREA_CRC32 for address range (800h - 83Fh)

Table 99. CMPA lifecycle area

CMPA lifecycle field	Offset	Bit	Description
LIFECYCLE_STATE	1000h	[31:0]	The field that advances lifecycle.
LIFECYCLE_STATE_DP	1004h	[31:0]	

19.2.3 Low-power wakeup path

The bootloader provides a fast low-power wakeup mechanism for Deep Power Down (DPD) mode.

When SRAM is retained and FAST_WAKEUP_EN in CMPA is enabled, the bootloader can execute low-power wakeup path after waking up from DPD mode. Bootloader performs normal boot if low-power wakeup path fails.

The following figure illustrates the low-power wakeup path.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

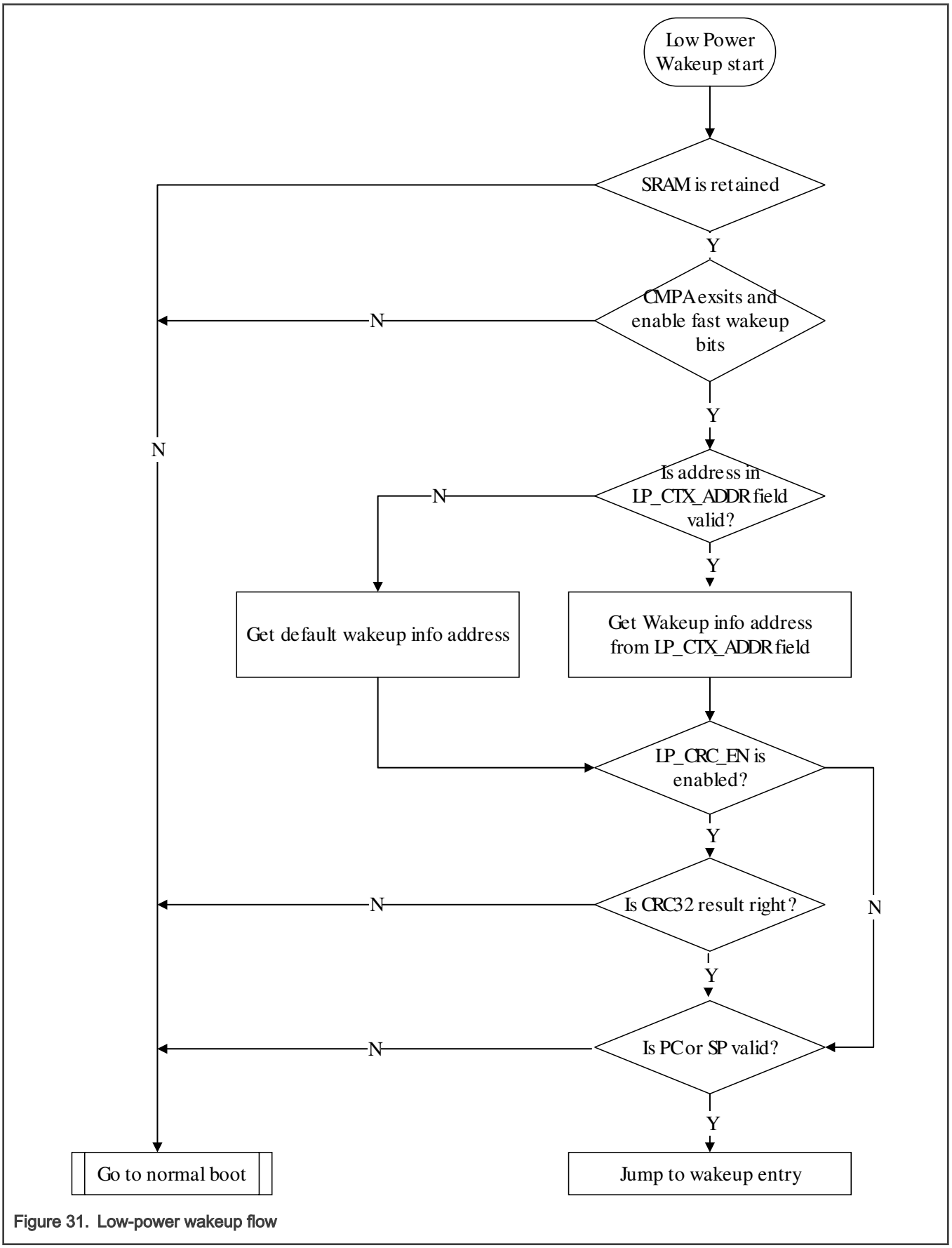


Figure 31. Low-power wakeup flow

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

When the device wakes up from Deep Power Down mode and executes low-power wakeup path, the bootloader can optionally jump to a wakeup routine instead of the standard boot image—provided that valid wakeup routine information is set in the specific SRAM address.

The bootloader will retrieve wakeup routine information from the LOW_POWER_CONTEXT_ADDRESS field of CMPA and will verify the validity of the SRAM address. If the address is invalid, the bootloader will obtain the information from the default SRAM address(0x2000_17E8). The wakeup routine information includes the following fields.

Table 100. Wakeup routine information

Wakeup routine information	offset	Description
PC	0x0	Valid WAKEUP Entry (PC).
SP	0x4	Stack Pointer (SP)
imgVertor	0x8	Vector Table address.
checkStart	0xC	Start address of data calculates CRC32 value.
checkLen	0x10	Length of data that calculates CRC32 value.
crcValue	0x14	Computed CRC32 result of data.

When the wakeup path is enabled and the system wakes up from Deep Power Down (DPD) mode, the bootloader enters the low-power wakeup path if both the WAKEUP entry point (PC) and the stack pointer (SP) are valid.

In the low-power wakeup flow, CRC checking is supported when LP_CRC_EN bits in CMPA are enabled. In this case, the bootloader reads the CRC check start address and length from the wakeup routine information and calculates the CRC32 value over this range. If the preloaded CRC value in the wakeup routine information matches the computed result, the bootloader transfers control to the wakeup routine at the entry point specified in the wakeup routine information. Otherwise, the bootloader exits the low-power wakeup path and proceeds with the normal boot sequence.

19.3 In-System Programming (ISP)

Bootloader provides in-system programming utility that operates over a serial connection on the MCUs. It enables quick and easy programming of MCUs to help application development and product manufacturing. Host-side tool is available to communicate with the bootloader. Users can utilize host tools to upload/download application code and do manufacturing via the bootloader.

When bootloader enters ISP mode, and it will auto-detect activity on the LPI2C/LPSPI or LPUART interface. The auto-detect looks for activity on the LPUART, LPI2C and LPSPI interface and selects the appropriate interface once a properly formed frame is received. If an invalid frame is received, the data is discarded and scanning resumes.

Users can make bootloader go to ISP path by pressing the BOOT_CONFIG pin when IFR0_ISP_PIN_ENTRY pin is enabled in CMPA. The default value of IFR0_ISP_PIN_ENTRY is configured as enable. In addition, Bootloader will go to ISP path when normal boot fails.

19.3.1 Pin assignments

Table below shows the pin assignments of ISP and is the default assignment used by the bootloader. However, they can be changed in the CMPA settings, see CMPA tables for more details.

Table 101. Pin assignments

	ISP peripheral pin	Port pin assignment
UART	LPUART_RX	P0_2
	LPUART_TX	P0_3
I2C	LPI2C_SDA	P1_8

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table 101. Pin assignments (continued)

	ISP peripheral pin	Port pin assignment
	LPI2C_SCL	P1_9
SPI	LPSPI_SDO	P1_0
	LPSPI_SCK	P1_1
	LPSPI_SDI	P1_2
	LPSPI_PCS	P1_3
ISP Pin	Pin to enter ISP mode	P3_29

19.3.2 Available ISP commands

ISP commands availability is controlled by lifecycle. Table below shows the ISP commands available for each lifecycle. Details of several commands can be found in sections below.

Command	Description	Develop	In-field1/2	In-field Locked
Reset	Reset the device	Available	Available	Available
get-property <tag>	Query about various properties and settings	Available	Available	Available
set-property <tag> <value>	Change properties or options in Bootloader	Available	Available	NA
flash-erase-region	Erase one or more sectors of the flash memory	Available	NA	NA
flash-erase-all [<memoryID>]	Erase all flash memory expect Bootloader region	Available	Available	NA
read-memory <addr> <byte_count> [<file>]	Read memory at specified address	Available	NA	NA
write-memory <addr> [<file> | {{<hex-data>}}]	Write memory at specified address from file or string of hex values **NOTE** When write to SRAM, make sure the length of file or hex-data is 4-byte aligned	Available	NA	NA
Execute <address> <arg> <stackpointer>	Jumps to code at the provided address and	Available	NA	NA

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Command	Description	Develop	In-field1/2	In-field Locked
	does not return to the ROM bootloader			
fill-memory <addr> <byte_count> <pattern> [word | short | byte]	Fill memory with pattern; pattern size can be word(default), short or byte	Available	Available	NA

19.3.2.1 Available properties

Table below shows the available properties (tag) for this device.

Table 102. Available properties

Tag ID	Tag	Description	Set-Property Supported
0x01	kPropertyTag_BootloaderVersion	Current version of the MCU bootloader.	No
0x02	kPropertyTag_AvailablePeripherals	A mask of the available peripherals.	No
0x03	kPropertyTag_FlashStartAddress	Starting address of the on-chip flash.	No
0x04	kPropertyTag_FlashSizeInBytes	Size of the on-chip flash.	No
0x05	kPropertyTag_FlashSectorSize	Size in bytes of one sector of program flash. This is the minimum erase size.	No
0x06	kPropertyTag_FlashBlockCount	Count of program flash.	No
0x07	kPropertyTag_AvailableCommands	Mask of the available commands.	No
0x08	kPropertyTag_CheckStatus	Status code from the last CRC check operation. Available only if the CRC check feature is supported.	No
0x0A	kPropertyTag_VerifyWrites	Verify Writes Flag - Boolean controlling whether the bootloader verifies writes to flash. • 0 - No verification is done. • 1 - Enables verification.	Yes
0x0B	kPropertyTag_MaxPacketSize	Maximum supported packet size for the currently active peripheral interface.	No
0x0C	kPropertyTag_ReservedRegions	Reserved Regions - List of memory regions reserved by the bootloader. Returned as value pairs (<start-address-ofregion>,<end-address-of-region>). If HasDataPhase flag is not set, response packet parameter count indicates number	No

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table 102. Available properties (continued)

Tag ID	Tag	Description	Set-Property Supported
		of pairs. If HasDataPhase flag is set, second parameter is the number of bytes in the data phase.	
0x0E	kPropertyTag_RAMStartAddress	Starting address of the on-chip RAM	No
0x0F	kPropertyTag_RAMSizeInBytes	Size in bytes of the on-chip RAM	No
0x11	kPropertyTag_SecurityState	Flash Security Enabled Flag. Return device lifecycle.	No
0x1C	kPropertyTag_IrqNotifierPin	IRQ notifier pin - The pin selected as nIRQ pin.	Yes
0x18	kPropertyTag_TargetVersion	Target version	No
0x20	kPropertyTag_CheckBootStatusRegister	Get boot state from register.	No

19.3.3 Special command

19.3.3.1 Fill-memory

The fill-memory command is specifically used in In-field 1/2 lifecycle to input the Flash-erase-all command password. The permissible address range for writing is 400E86B8h to 400E86C7h(ERASE_PASSWORD_ADDRESS).

19.3.3.2 Flash-erase-all

The flash-erase-all command erases the user PFlash regions (offsets 0x0000–0xFFFF and 0x1C000–0x1FFFF) and IFR0 sectors 1 through 3. In the Develop lifecycle state, this command can be executed without restrictions.

```
blhost -p COMx -- flash-erase-all
```

In the In-field1 and In-field2 lifecycle states, a Power-On Reset (POR) must be performed before executing the mass erase sequence. The user must then use the fill-memory command to write four consecutive 32-bit password words to the starting address 0x400E86B8 (ERASE_PASSWORD_ADDRESS). If the written values match the ERASE_PASSWORD words stored in the CMPA (offset 0x830), the flash-erase-all command is permitted.

NOTE

If the image was booted during In-field 1/2 lifecycle, a POR reset must be performed, followed by direct entry into the ISP and then perform flash-erase-all command.

```
blhost -p COMx -- fill-memory 400E86B8h 4 ERASE_PASSWORD1
blhost -p COMx -- fill-memory 400E86BCh 4 ERASE_PASSWORD2
blhost -p COMx -- fill-memory 400E86C0h 4 ERASE_PASSWORD3
blhost -p COMx -- fill-memory 400E86C4h 4 ERASE_PASSWORD4
blhost -p COMx -- Flash-erase-all
```

19.3.4 In-System Programming protocol

This section explains the general protocol for the packet transfers between the host and the Bootloader. The description includes the transfer of packets for different transactions, such as commands with no data phase and commands with incoming or outgoing data phase. The next section describes various packet types used in a transaction.

Each command sent from the host is replied to with a response command.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Commands may include an optional data phase.

- If the data phase is incoming (from the host to the bootloader), it is part of the original command.
- If the data phase is outgoing (from the bootloader to host), it is part of the response command.

19.3.4.1 Command with no data phase

The protocol for a command with no data phase contains:

- Command packet (from the host)
- Generic response command packet (to host).

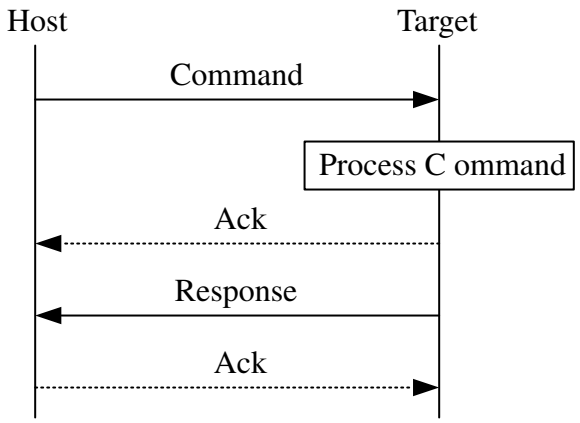


Figure 32. Command with no data phase

NOTE

In these diagrams, the ACK sent in response to a command, or a data packet can arrive at any time before, during, or after the command or data packet has processed.

19.3.4.2 Command with incoming data phase

The protocol for a command with incoming data phase contains:

- Command packet (from host) (kCommandFlag_HasDataPhase set).
- Generic response command packet (to host).
- Incoming data packets (from the host).
- Generic response command packet.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

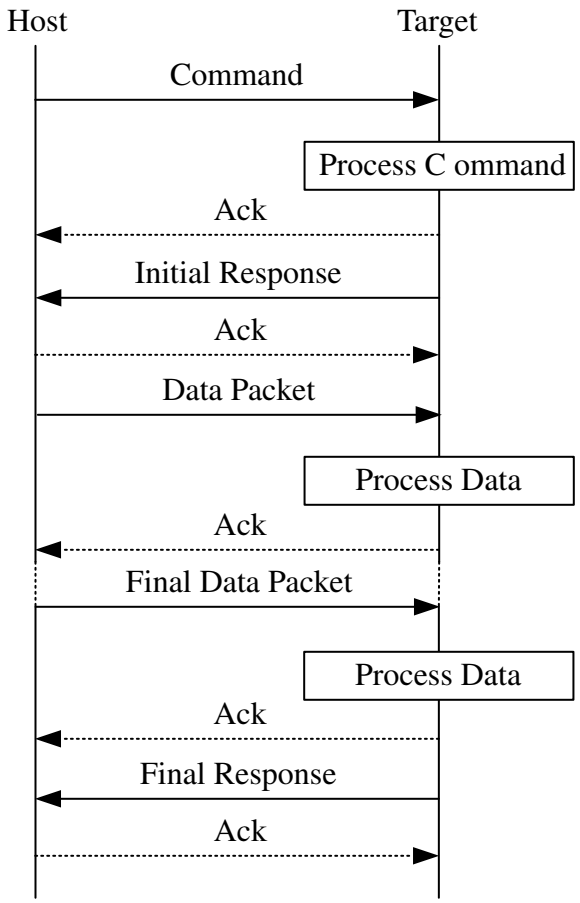


Figure 33. Command with incoming data phase

NOTE

- The host may not send any further packets while it is waiting for the response to a command.
- The data phase is aborted if the Generic Response packet prior to the start of the Data phase does not have a status of kStatus_Success.
- Data phases may be aborted by the receiving side by sending the final GenericResponse early with a status of kStatus_AbortDataPhase. The host may abort the data phase early by sending a zero-length data packet.
- The final Generic Response packet sent after the data phase includes the status of the entire operation.

19.3.4.3 Command with outgoing data phase

The protocol for a command with an outgoing data phase contains:

- Command packet (from the host).
- ReadMemory Response command packet (to host) (kCommandFlag_HasDataPhase set).
- Outgoing data packets (to host).
- Generic response command packet (to host).

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

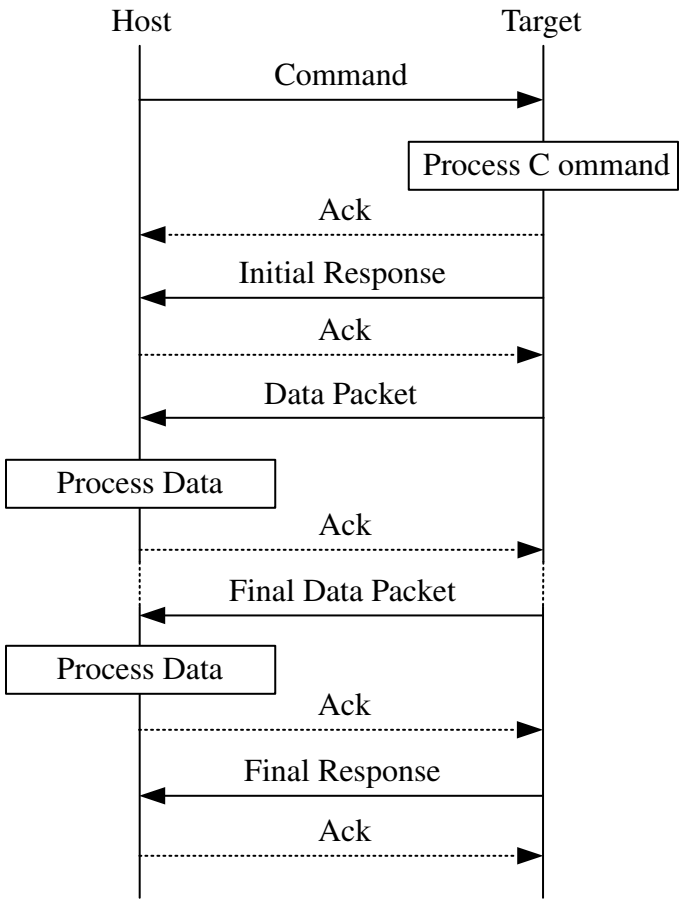


Figure 34. Command with outgoing data phase

NOTE

- The data phase is considered part of the response command for the outgoing data phase sequence. The host may not send any further packets while the host is waiting for the response to a command.
- The data phase is aborted if the ReadMemory Response command packet, prior to the start of the data phase, does not contain the kCommandFlag_HasDataPhase flag. Data phases may be aborted by the host sending the final Generic Response early with a status of kStatus_AbortDataPhase. The sending side may abort the data phase early by sending a zero-length data packet.
- The final Generic Response packet sent after the data phase includes the status of the entire operation.

19.3.5 ISP packet type

The bootloader device works in slave mode. All data communications are initiated by a host, which is either a PC or an embedded host. The bootloader device is the target, which receives a command or data packet. All data communications between host and target are packetized.

There are six types of packets used:

- Ping packet.
- Ping Response packet.
- Framing packet.
- Command packet.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

- Data packet.
- Response packet.

All fields in the packets are in little-endian byte order.

19.3.5.1 Ping packet

The Ping packet is the first packet sent from a host to the target to establish a connection on the selected peripheral in order to run autobaud detection. The Ping packet can be sent from host to target at any time that the target is expecting a command packet. If the selected peripheral is LPUART, a ping packet must be sent before any other communications. For other serial peripherals, it is optional.

In response to a Ping packet, the target sends a Ping Response packet, discussed in later sections.

Table 103. Ping packet format

Byte #	Value	Name
0	5Ah	Start byte
1	6Ah	Ping

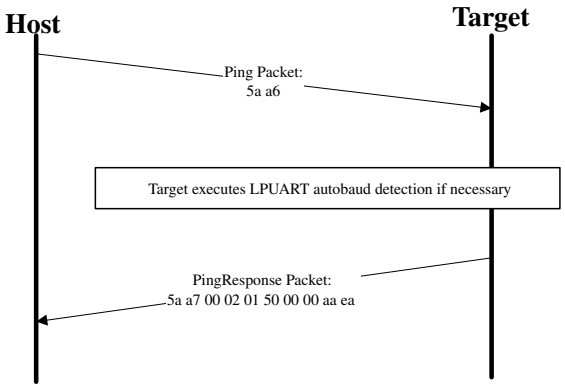


Figure 35. Ping packet protocol sequence

19.3.5.2 Ping response packet

The target sends a Ping Response packet back to the host after receiving a Ping packet. If communication is over an LPUART peripheral, the target uses the incoming Ping packet to determine the baud rate before replying with the Ping Response packet. Once the Ping Response packet is received by the host, the connection is established, and the host starts sending commands to the target.

Byte #	Value	Parameter
0	5Ah	Start byte
1	A7h	Ping response code
2	00h	Protocol bugfix
3	03h	Protocol minor
4	01h	Protocol major
5	50h	Protocol name = 'P' (50h)

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Byte #	Value	Parameter
6	00h	Options low
7	00h	Options high
8	aah	CRC16 low
9	eah	CRC16 high

The Ping Response packet can be sent from host to target any time the target expects a command packet. For the LPUART peripheral, it must be sent by the host when a connection is first established, in order to run outbound. For other serial peripherals, it is optional but recommended to determine the serial protocol version. The version number is in the same format as the bootloader version number returned by the GetProperty command.

19.3.5.3 Framing packet

The framing packet is used for flow control and error detection for the communications links that do not have such features built-in. The framing packet structure sits between the link layer and the command layer. It wraps command and data packets as well.

Every framing packet containing data sent in one direction results in a synchronizing response framing packet in the opposite direction.

The framing packet described in this section is used for serial peripherals including the LPUART, LPI2C, and LPSPI.

Byte #	Value	Parameter	Remark
0	5Ah	Start byte	
1		Packet type	
2		Length low	Length is a 16-bit field that specifies the entire command or data packet size in bytes.
3		Length high	
4		CRC16 low	This is a 16-bit field. The CRC16 value covers the entire framing packet, including the start byte and command or data packets, but does not include the CRC bytes. See the CRC16 algorithm after this table.
5		CRC16 high	
6...n		Command or data packet payload	

A special framing packet that contains only a start byte and a packet type is used for synchronization between the host and target.

Table 104. Special framing packet format

Byte #	Value	Parameter
0	5Ah	Start byte
1	Ahn	Packet type

The Packet Type field specifies the type of the packet from one of the defined types (below):

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Packet type	Name	Description
A1h	kFramingPacketType_Ack	The previous packet was received successfully; the sending of more packets is allowed.
A2h	kFramingPacketType_Nak	The previous packet was corrupted and must be re-sent.
A3h	kFramingPacketType_AckAbort	Data phase is being aborted.
A4h	kFramingPacketType_Command	The framing packet contains a command packet payload.
A5h	kFramingPacketType_Data	The framing packet contains a data packet payload.
A6h	kFramingPacketType_Ping	Sent to verify the other side is alive. Also used for LPUART autobaud.
A7h	kFramingPacketType_PingResponse	A response to Ping; contains the framing protocol version number and options.

19.3.5.4 CRC16 algorithm

This section provides the CRC16 algorithm.

The CRC is computed over each byte in the framing packet header, excluding the crc16 field itself, plus all of the payload bytes. The CRC algorithm is the XMODEM variant of CRC-16.

Table 105. Characteristics of the XMODEM variant

Variant	Value
Width	16
Polynomial	1021h
Init value	0h
Reflect in	False
Reflect out	False
Xor out	0h
Check result	31c3h

The check result is computed by running the ASCII character sequence "123456789" through the algorithm.

```
uint16_t crc16_update(const uint8_t * src, uint32_t lengthInBytes)
{
    uint32_t crc = 0;
    uint32_t j;

    for (j = 0; j < lengthInBytes; ++j)
    {
        uint32_t i;
        uint32_t byte = src[j];

        crc ^= (byte << 8);
```


All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

```
        for (i = 0; i < 8; ++i)
        {
            uint32_t temp = (crc << 1);
            if (crc & 8000h)
            {
                temp ^= 1021h;
            }
            crc = temp;
        }
    }

    return (uint16_t)crc;
}
```

19.3.5.5 Command packet

The command packet carries a 32-bit command header and a list of 32-bit parameters.

Table 106. Command packet format (32 bytes)

Command header (4 bytes)				28 bytes for Parameters (Max 7 parameters)						
Tag	Flags	Reserved	Param Count	Param1 (32-bit)	Param2 (32-bit)	Param3 (32-bit)	Param4 (32-bit)	Param5 (32-bit)	Param6 (32-bit)	Param7 (32-bit)
Byte 0	Byte 1	Byte 2	Byte 3	-	-	-	-	-	-	-

Table 107. Command header format

Byte #	Command header field	Remark
0	Command or Response tag	The command header is 4 bytes long, with these fields.
1	Flags	
2	Reserved. Should be 00h.	
3	ParameterCount	

The header is followed by 32-bit parameters up to the value of the ParameterCount field specified in the header. Because a command packet is 32 bytes long, only 7 parameters can fit into the command packet.

Command packets are also used by the target to send responses back to the host. As mentioned earlier, command packets and data packets are embedded into framing packets for all the transfers.

Table 108. Command tags

Command Tag	Name	Remark
01h	FlashEraseAll	The command tag specifies one of the commands supported by the bootloader. The valid command tags for the bootloader are listed here.
02h	FlashEraseRegion	
03h	ReadMemory	
04h	WriteMemory	
05h	FillMemory	
07h	GetProperty	

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table 108. Command tags (continued)

Command Tag	Name	Remark
09h	Execute	
0Bh	Reset	
0Ch	SetProperty	
Other value	Reserved	

Table 109. Response tags

Response tag	Name	Remark
A0h	GenericResponse	The response tag specifies one of the responses the bootloader (target) returns to the host. The valid response tags are listed here.
A3h	ReadMemoryResponse	
A7h	GetPropertyResponse	

Flags: Each command packet contains a Flag byte. Only bit 0 of the flag byte is used. If bit 0 of the flag byte is set to 1, then data packets follow the command sequence. The number of bytes that are transferred in the data phase is determined by a command specific parameter in the parameters array.

ParameterCount: The number of parameters included in the command packet.

Parameters: The parameters are word-length (32 bits). With the default maximum packet size of 32 bytes, a command packet can contain up to 7 parameters.

19.3.5.6 Response packet

The responses are carried using the same command packet format wrapped with framing packet data. Types of responses include:

- GenericResponse
- GetPropertyResponse
- ReadMemoryResponse
- FlashReadOnceResponse
- KeyProvisionResponse

GenericResponse: After the bootloader has processed a command, the bootloader sends a generic response with status and command tag information to the host. The generic response is the last packet in the command protocol sequence. The generic response packet contains the framing packet data and the command packet data (with generic response tag = A0h) and a list of parameters (defined in the next section). The parameter count field in the header is always set to 2, for status code and command tag parameters.

Table 110. GenericResponse parameters

Byte #	Parameter	Description
0-3	Status code	The Status codes are errors encountered during the execution of a command by the target. If a command succeeds, then a kStatus_Success code is returned.

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table 110. GenericResponse parameters (continued)

Byte #	Parameter	Description
4-7	Command tag	The Command tag parameter identifies the response to the command sent by the host.

GetPropertyResponse: The GetPropertyResponse packet is sent by the target in response to the host query that uses the GetProperty command. The GetPropertyResponse packet contains the framing packet data and the command packet data, with the command/response tag set to a GetPropertyResponse tag value (A7h).

The parameter count field in the header is set to greater than 1, to always include the status code and one or many property values

Table 111. GetPropertyResponse parameters

Byte #	Parameter
0-3	Status code
4-7	Property value
...	...
	Can be up to maximum 6 property values, limited to the size of the 32-bit command packet and property type.

ReadMemoryResponse: The ReadMemoryResponse packet is sent by the target in response to the host sending a ReadMemory command. The ReadMemoryResponse packet contains the framing packet data and the command packet data, with the command/response tag set to a ReadMemoryResponse tag value (A3h), the flags field set to kCommandFlag_HasDataPhase (1).

The parameter count set to 2 for the status code and the data byte count parameters shown below.

Table 112. ReadMemoryResponse parameters

Byte #	Parameter	Description
0-3	Status code	The status of the associated Read Memory command.
4-7	Data byte count	The number of bytes sent in the data phase.

FlashReadOnceResponse: The FlashReadOnceResponse packet is sent by the target in response to the host sending a FlashReadOnce command. The FlashReadOnceResponse packet contains the framing packet data and the command packet data, with the command/response tag set to a FlashReadOnceResponse tag value (AFh), and the flags field set to 0. The parameter count is set to 2 plus the number of words requested to be read in the FlashReadOnceCommand.

Table 113. FlashReadOnceResponse parameters

Byte #	Parameter
0-3	Status code
4-7	Byte count to read
...	...
	Can be up to 20 bytes of requested read data.

The KeyProvisionResponse packet is sent by the target in response to the host sending a KeyProvision command. The KeyProvisionResponse packet contains the framing packet data and command packet data, with the command/response tag set to a KeyProvisionResponse tag value (B5h), and the flags field set to kCommandFlag_HasDataPhase (1).

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table 114. KeyProvisionResponse parameters

Byte #	Parameter
0-3	Status code
4-7	Data btye count

19.3.6 Bootloader command set

All bootloader commands follow the command packet format wrapped by the framing packet as explained in previous sections.

See Available ISP commands for different lifecycle for a list of commands supported by the bootloader.

19.3.6.1 GetProperty command

The GetProperty command is used to query the bootloader about various properties and settings. Each supported property has a unique 32-bit tag associated with it. The tag occupies the first parameter of the command packet. The target returns a GetPropertyResponse packet with the property values for the property identified with the tag in the GetProperty command.

Properties are the defined units of data that can be accessed with the GetProperty or SetProperty commands. Properties may be read-only or read-write. All read-write properties are 32-bit integers, so they can easily be carried in a command parameter.

The 32-bit property tag is the only parameter required for GetProperty command.

Table 115. Parameters for GetProperty command

Byte #	Command
0-3	Property tag See Available properties for more details.
4-7	External Memory Identifier (only applies to get property for external memory)

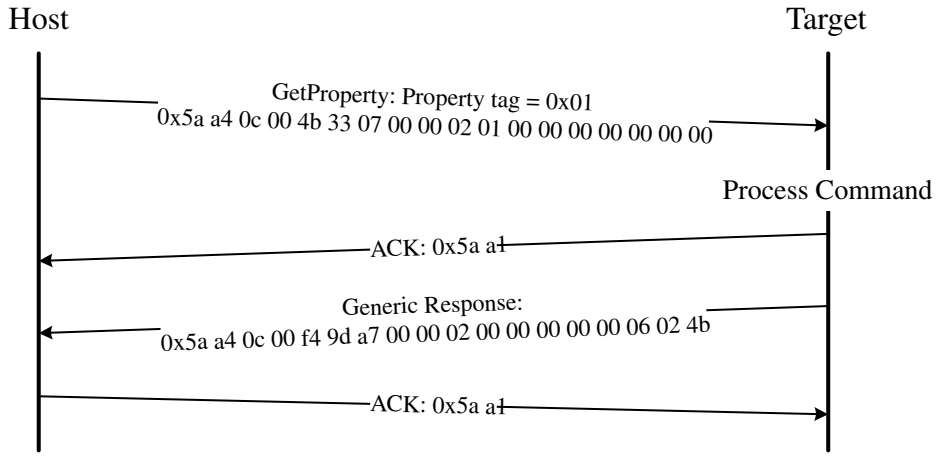


Figure 36. Protocol sequence for GetProperty command

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table 116. GetProperty command packet format (example)

GetProperty	Parameter	Value
Framing packet	start byte	5Ah
	packetType	A4h, kFramingPacketType_Command
	length	0Ch 00h
	crc16	4Bh 33h
Command packet	commandTag	07h - GetProperty
	flags	00h
	reserved	00h
	parameterCount	02h
	propertyTag	00000001h - CurrentVersion
	Memory ID	00000000h - Internal Flash

The GetProperty command has no data phase.

Response: In response to a GetProperty command, the target sends a GetPropertyResponse packet with the response tag set to A7h. The parameter count indicates the number of parameters sent for the property values, with the first parameter showing status code 0, followed by the property value(s). The next table shows an example of a GetPropertyResponse packet.

Table 117. GetProperty response packet format (example)

GetPropertyResponse	Parameter	Value
Framing packet	start byte	5Ah
	packetType	A4h, kFramingPacketType_Command
	length	0ch 00h (12 bytes)
	crc16	07h 7ah
Command packet	responseTag	A7h
	flags	00h
	reserved	00h
	parameterCount	02h
	status	00000000h
	propertyValue	00000014h - CurrentVersion

19.3.6.2 SetProperty command

The SetProperty command is used to change or alter the values of the properties or options of the bootloader. The command accepts the same property tags used with the GetProperty command. However, only some properties are writable--see Available properties. If an attempt to write a read-only property is made, an error is returned indicating the property is read-only and cannot be changed.

The property tag and the new value to set are the two parameters required for the SetProperty command.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table 118. Parameters for SetProperty command

Byte #	Command
0-3	Property tag See Available properties for more details.
4-7	Property value

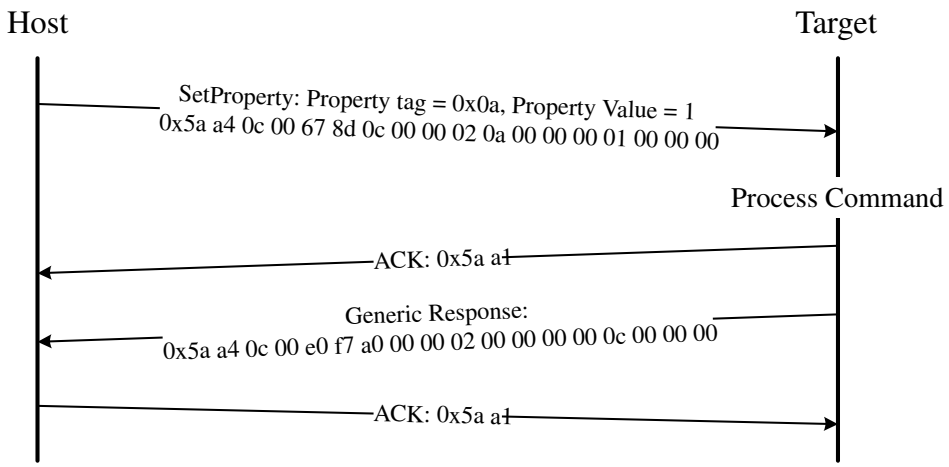


Figure 37. Parameters for SetProperty command

Table 119. SetProperty command packet format (example)

SetProperty	Parameter	Value
Framing packet	start byte	5Ah
	packetType	A4h, kFramingPacketType_Command
	length	0Ch 00h
	crc16	67h 8Dh
Command packet	commandTag	0Ch - SetProperty with property tag 10
	flags	00h
	reserved	00h
	parameterCount	02h
	propertyTag	0000000Ah - VerifyWrites
	propertyValue	00000001h

The SetProperty command has no data phase.

Response: The target returns a GenericResponse packet with one of following status codes.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table 120. SetProperty response status codes

Status code
kStatus_Success
kStatus_ReadOnly
kStatus_UnknownProperty
kStatus_InvalidArgument

19.3.6.3 FlashEraseAll command

The FlashEraseAll command performs an erase of the entire flash memory. If any flash regions are protected, then the FlashEraseAll command fails and returns an error status code. The Command tag for FlashEraseAll command is 01h set in the commandTag field of the command packet.

Table 121. Parameter for FlashEraseAll command

Byte #	Parameter	
	Memory ID	Descriptions
0–3	000h	Internal Flash
	002h	Radio PFlash
	003h	Radio User IFR

The FlashEraseAll command performs an erase of the entire flash memory. If any flash regions are protected, then the FlashEraseAll command fails and returns an error status code. The Command tag for FlashEraseAll command is 01h set in the commandTag field of the command packet. See special command for more details.

Table 122. Parameter for FlashEraseAll command

Byte #	Parameter	
	Memory ID	Descriptions
0–3	000h	Internal Flash
	002h	Radio PFlash
	003h	Radio User IFR

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

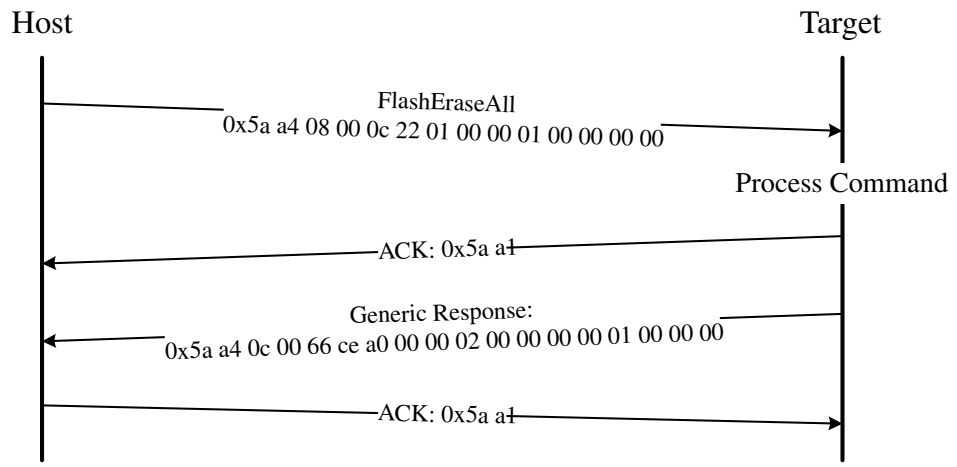


Figure 38. Protocol sequence for FlashEraseAll command

Table 123. FlashEraseAll command packet format (example)

FlashEraseAll	Parameter	Value
Framing packet	start byte	5Ah
	packetType	A4h, kFramingPacketType_Command
	length	08h 00h
	crc16	0Ch 22h
Command packet	commandTag	01h- FlashEraseAll
	flags	00h
	reserved	00h
	parameterCount	01h
	Memory ID	See the above table

The FlashEraseAll command has no data phase.

Response: The target returns a GenericResponse packet with status code either set to kStatus_Success for successful execution of the command or set to an appropriate error status code.

19.3.6.4 FlashEraseRegion command

The FlashEraseRegion command performs an erase of one or more sectors of the flash memory.

The start address, and number of bytes are the 2 parameters required for the FlashEraseRegion command. The start and byte count parameters must be 32-byte aligned ([3:0] = 0000), or the FlashEraseRegion command fails and returns kStatus_FlashAlignmentError (101). If the region specified does not fit in the flash memory space, the FlashEraseRegion command fails and returns kStatus_FlashAddressError (102).

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

If any part of the region specified is protected, the FlashEraseRegion command fails and returns kStatus_MemoryRangeInvalid (10200).

Table 124. Parameters for FlashEraseRegion command

Byte #	Parameter
0 - 3	Start address
4 - 7	Byte count
8 - 11	Memory ID

The FlashEraseRegion command has no data phase.

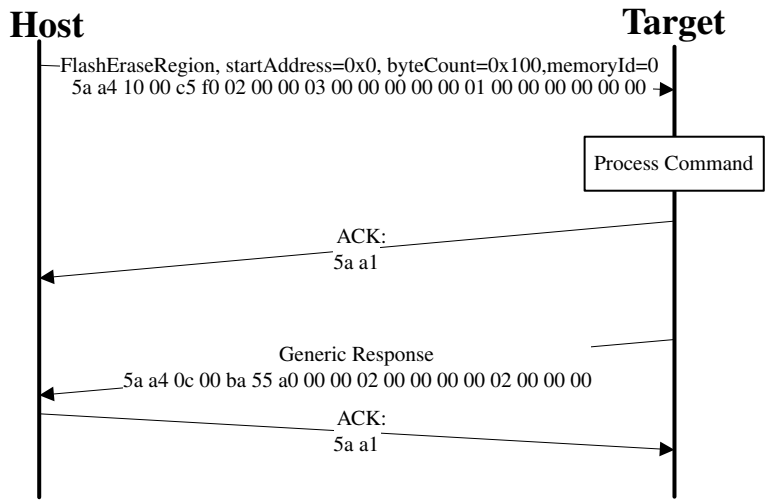


Figure 39. Protocol sequence for FlashEraseRegion command

Response: The target returns a GenericResponse packet with one of following error status codes.

Status code
kStatus_Success (0)
kStatus_MemoryRangeInvalid (10200)
kStatus_FlashAlignmentError (101)
kStatus_FlashAddressError (102)
kStatus_FlashAccessError (103)
kStatus_FlashCommandFailure (105)

19.3.6.5 ReadMemory command

The ReadMemory command returns the contents of memory at the given address, for a specified number of bytes. This command can read any region of memory accessible by the CPU and not protected by security.

The start address, and number of bytes are the two parameters required for ReadMemory command. The memory ID is optional. Internal memory will be selected as default if memory ID is not specified.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table 125. Parameters for read memory command

Byte	Parameter	Description
0 - 3	Start address	Start address of memory to read from
4 - 7	Byte count	Number of bytes to read and return to caller
8 - 11	Memory ID	Internal or external memory identifier

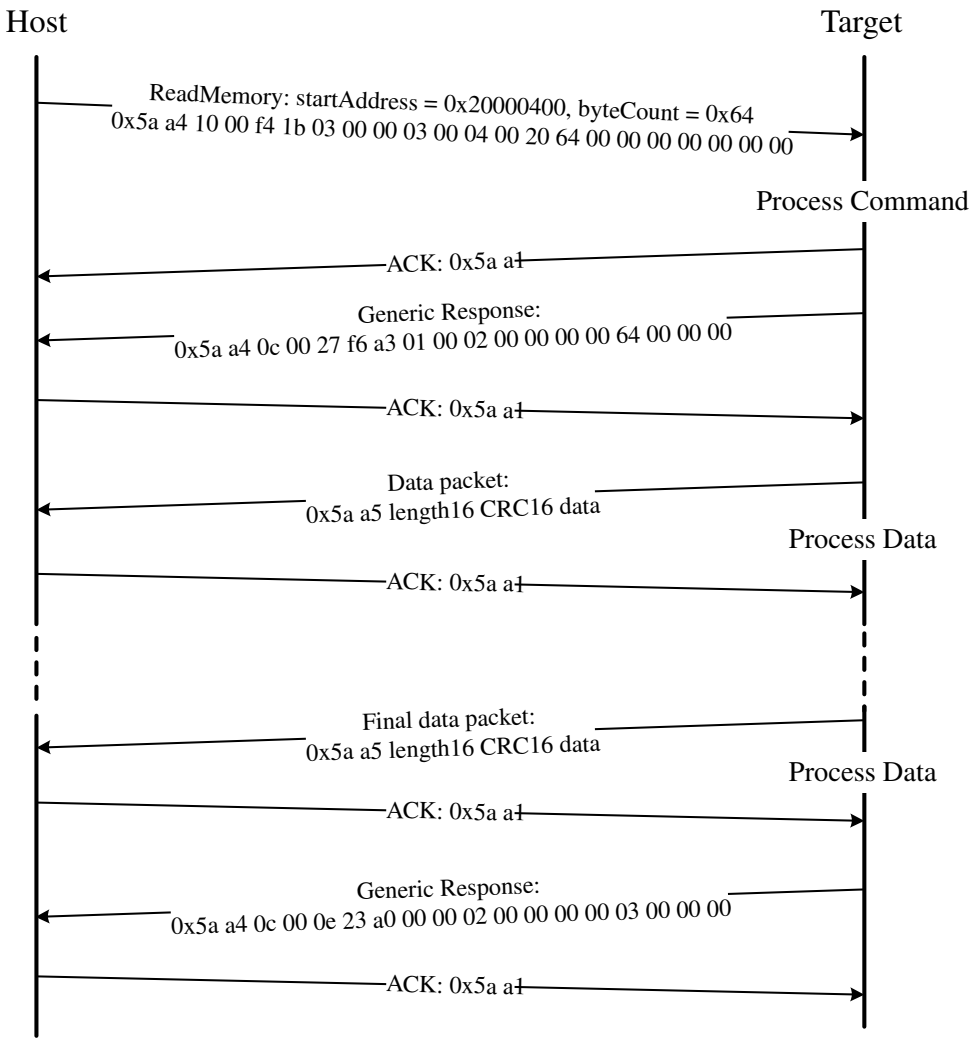


Figure 40. Command sequence for ReadMemory

Data phase: The ReadMemory command has a data phase. Because the target works in slave mode, the host needs to pull data packets until the number of bytes of data specified in the byteCount parameter of the ReadMemory command are received by the host.

Response: The target returns a GenericResponse packet with a status code either set to kStatus_Success upon successful execution of the command or set to an appropriate error status code.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

19.3.6.6 WriteMemory command

The WriteMemory command writes data provided in the data phase to a specified range of bytes in memory (flash or RAM). However, if flash protection is enabled, then writes to protected sectors fail.

Special care must be taken when writing to flash.

- First, any flash sector written to must have been previously erased with a FlashEraseAll or FlashEraseRegion.
- First, any flash sector written to must have been previously erased with a FlashEraseAll or FlashEraseRegion command.
- Writing to flash requires the start address to be 16-byte aligned ([3:0] = 0000).
- The byte count is rounded up to a multiple of 4, and trailing bytes are filled with the flash erase pattern (ffh).
- If the VerifyWrites property is set to true, then writes to flash also performs a flash verify program operation.

When writing to RAM, the start address does not need to be aligned, and the data is not padded.

The start address and number of bytes are the 2 parameters required for WriteMemory command. The memory ID is optional. Internal memory will be selected as default if memory ID is not specified.

Table 126. Parameters for WriteMemory command

Byte #	Command
0 - 3	Start address
4 - 7	Byte count
8 - 11	Memory ID

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

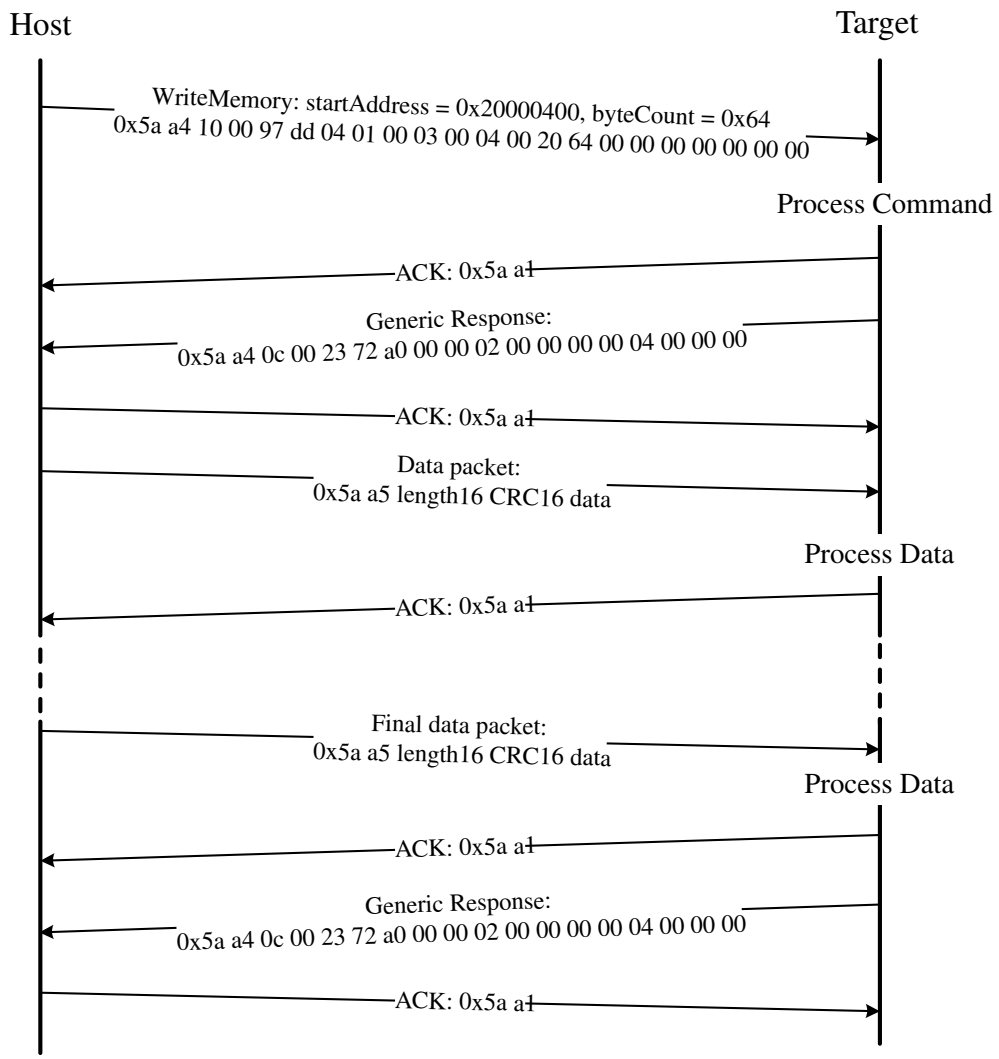


Figure 41. Protocol sequence for WriteMemory command

Table 127. WriteMemory command packet format (example)

WriteMemory	Parameter	Value
Framing packet	start byte	5Ah
	packetType	A4h, kFramingPacketType_Command
	length	10h 00h
	crc16	97h DDh
Command packet	commandTag	04h - WriteMemory

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table 127. WriteMemory command packet format (example) (continued)

WriteMemory	Parameter	Value
	flags	01h
	reserved	00h
	parameterCount	03h
	startAddress	20000400h
	byteCount	00000064h
	memoryID	0h

Data phase: The WriteMemory command has a data phase; the host sends data packets until the number of bytes of data specified in the byteCount parameter of the WriteMemory command are received by the target.

Response: The target returns a GenericResponse packet with a status code set to kStatus_Success upon successful execution of the command, or to an appropriate error status code.

19.3.6.7 FillMemory command

The FillMemory command fills a range of bytes in memory with a data pattern. It follows the same rules as the WriteMemory command. The difference between FillMemory and WriteMemory is that a data pattern is included in FillMemory command parameter, and there is no data phase for the FillMemory command, while WriteMemory does have a data phase.

Table 128. Parameters for FillMemory command

Byte #	Command
0 - 3	Start address of memory to fill
4 - 7	Number of bytes to write with the pattern • The start address should be 32-bit aligned. • The number of bytes must be evenly divisible by 4.
8 - 11	32-bit pattern

- To fill with a byte pattern (8-bit), the byte must be replicated 4 times in the 32-bit pattern.
- To fill with a short pattern (16-bit), the short value must be replicated 2 times in the 32-bit pattern.

For example, to fill a byte value with FEh, the word pattern is FEFEFEFEh; to fill a short value 5AFEh, the word pattern is 5AFE5AFEh.

Special care must be taken when writing to flash.

- First, any flash sector written to must have been previously erased with a FlashEraseAll, or FlashEraseRegion command.
- First, any flash sector written to must have been previously erased with a FlashEraseAll or FlashEraseRegion command.
- Writing to flash requires the start address to be 16-byte aligned ([3:0] = 0000).
- If the VerifyWrites property is set to true, then writes to flash also performs a flash verify program operation.

When writing to RAM, the start address does not need to be aligned, and the data is not padded.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

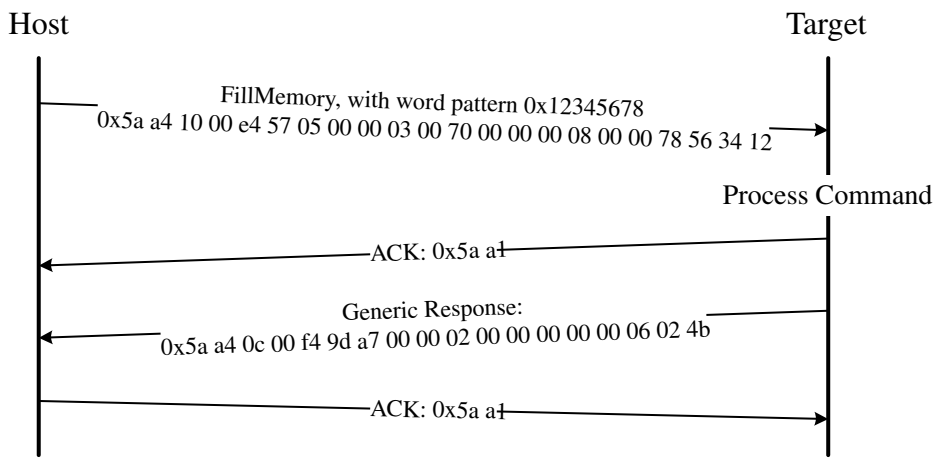


Figure 42. Protocol sequence for FillMemory command

Table 129. FillMemory command packet format (example)

FillMemory	Parameter	Value
Framing packet	start byte	5Ah
	packetType	A4h, kFramingPacketType_Command
	length	10h 00h
	crc16	E4h 57h
Command packet	commandTag	05h - FillMemory
	flags	00h
	reserved	00h
	parameterCount	03h
	startAddress	00007000h
	byteCount	00000800h
	patternWord	12345678h

The FillMemory command has no data phase.

Response: Upon successful execution of the command, the target (Kinetis bootloader) returns a GenericResponse packet with a status code set to kStatus_Success, or to an appropriate error status code.

19.3.6.8 Execute command

The execute command results in the bootloader setting the program counter to the code at the provided jump address, R0 to the provided argument, and a Stack pointer to the provided stack pointer address. Prior to the jump, the system is returned to the reset state.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

The Jump address, function argument pointer, and stack pointer are the parameters required for the Execute command. If the stack pointer is set to zero, the called code is responsible for setting the processor stack pointer before using the stack.

Table 130. Parameters for execute command

Byte #	Command
0 - 3	Jump address
4 - 7	Argument word
8 - 11	Stack pointer address

The Execute command has no data phase.

Response: Before executing the Execute command, the target validates the parameters and returns a GenericResponse packet with a status code either set to kStatus_Success or an appropriate error status code.

19.3.6.9 Reset command

The Reset command results in the bootloader resetting the chip.

The Reset command requires no parameters.

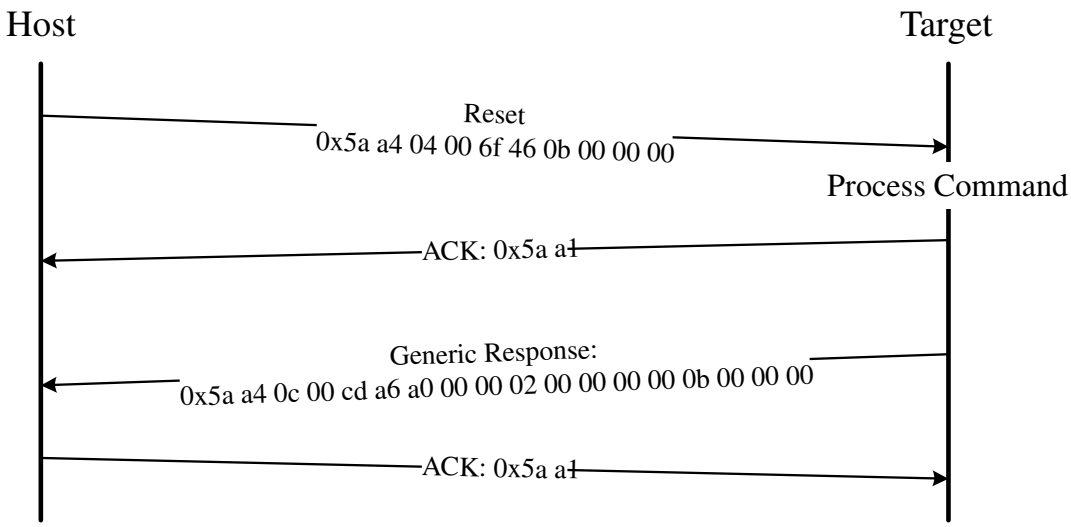


Figure 43. Protocol sequence for Reset command

Table 131. Reset command packet format (example)

Reset	Parameter	Value
Framing packet	start byte	5Ah
	packetType	A4h, kFramingPacketType_Command
	length	04h 00h

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table 131. Reset command packet format (example) (continued)

Reset	Parameter	Value
	crc16	6Fh 46h
Command packet	commandTag	0Bh - reset
	flags	00h
	reserved	00h
	parameterCount	00h

The Reset command has no data phase.

Response: The target returns a GenericResponse packet with status code set to kStatus_Success, before resetting the chip.

The reset command can also be used to switch boot from flash after successful flash image provisioning via ROM bootloader. After issuing the reset command, allow 5 seconds for the user application to start running from Flash.

19.3.7 LPUART ISP

The bootloader integrates an autobaud detection algorithm for the LPUART peripheral, thereby providing flexible baud rate choices.

Autobaud feature:If LPUARTn is used to connect to the bootloader, then the LPUARTn_RX pin must be kept high and not left floating during the detection phase in order to comply with the autobaud detection algorithm. After the bootloader detects the ping packet (5Ah A6h) on LPUARTn_RX, the bootloader firmware executes the autobaud sequence.

If the baudrate is successfully detected, then the bootloader sends a ping packet response [(5Ah A7h), protocol version (4 bytes), protocol version options (2 bytes) and crc16 (2 bytes)] at the detected baudrate. The bootloader then enters a loop, waiting for bootloader commands via the LPUART peripheral.

The data bytes of the ping packet must be sent continuously (with no more than 80 ms between bytes) in a fixed LPUART transmission mode (8-bit data, no parity bit and 1 stop bit). If the bytes of the ping packet are sent one-by-one with more than 80 ms delay between them, then the autobaud detection algorithm may calculate an incorrect baud rate. In this instance, the autobaud detection state machine should be reset.

NOTE

The data bytes of the ping packet must be sent continuously (with no more than 80 ms between bytes) in a fixed LPUART transmission mode (8-bit data, no parity bit and 1 stop bit). If the bytes of the ping packet are sent one-by-one with more than 80 ms delay between them, then the autobaud detection algorithm may calculate an incorrect baud rate. In this instance, the autobaud detection state machine should be reset.

Supported baud rates : The baud rate is closely related to the MCU core and system clock frequencies. Typical baud rates supported are 9600, 19200, 38400, 57600, 115200 and 230400.

Packet transfer : After autobaud detection succeeds, bootloader communications can take place over the LPUART peripheral. The following flow charts show:

- How the host detects an ACK from the target
- How the host detects a ping response from the target
- How the host detects a command response from the target

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

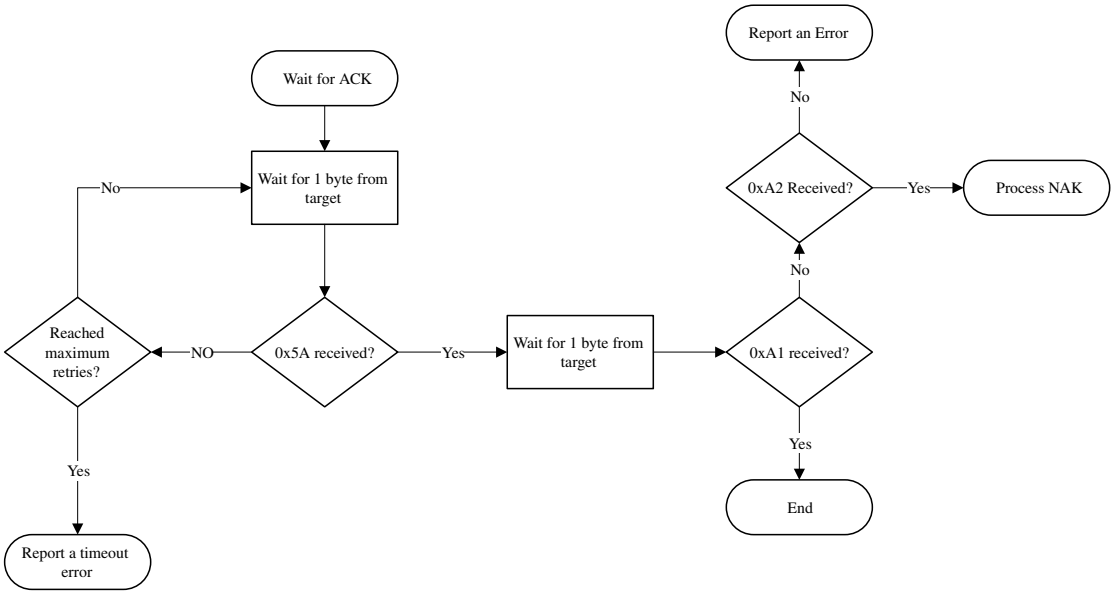


Figure 44. Host reads an ACK from target via LPUART

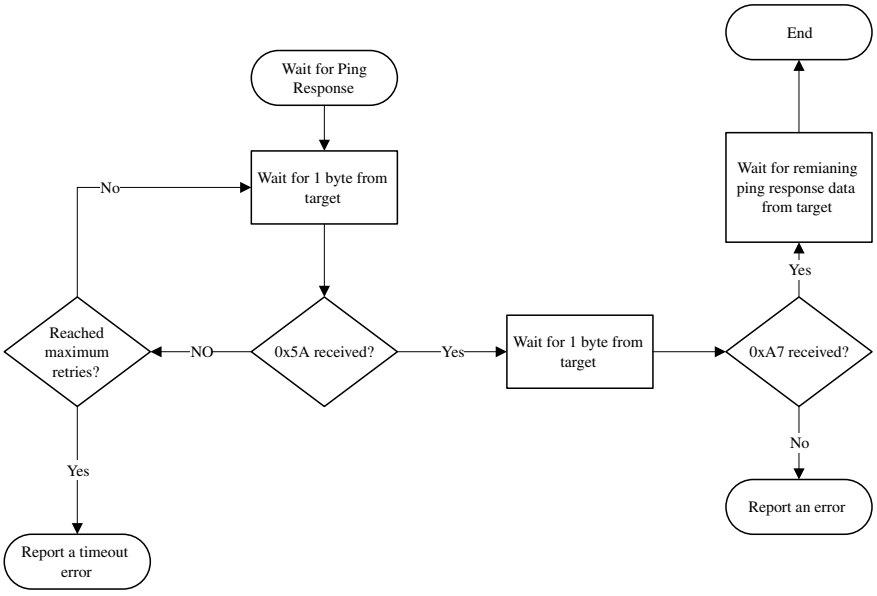


Figure 45. Host reads a ping response from target via LPUART

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

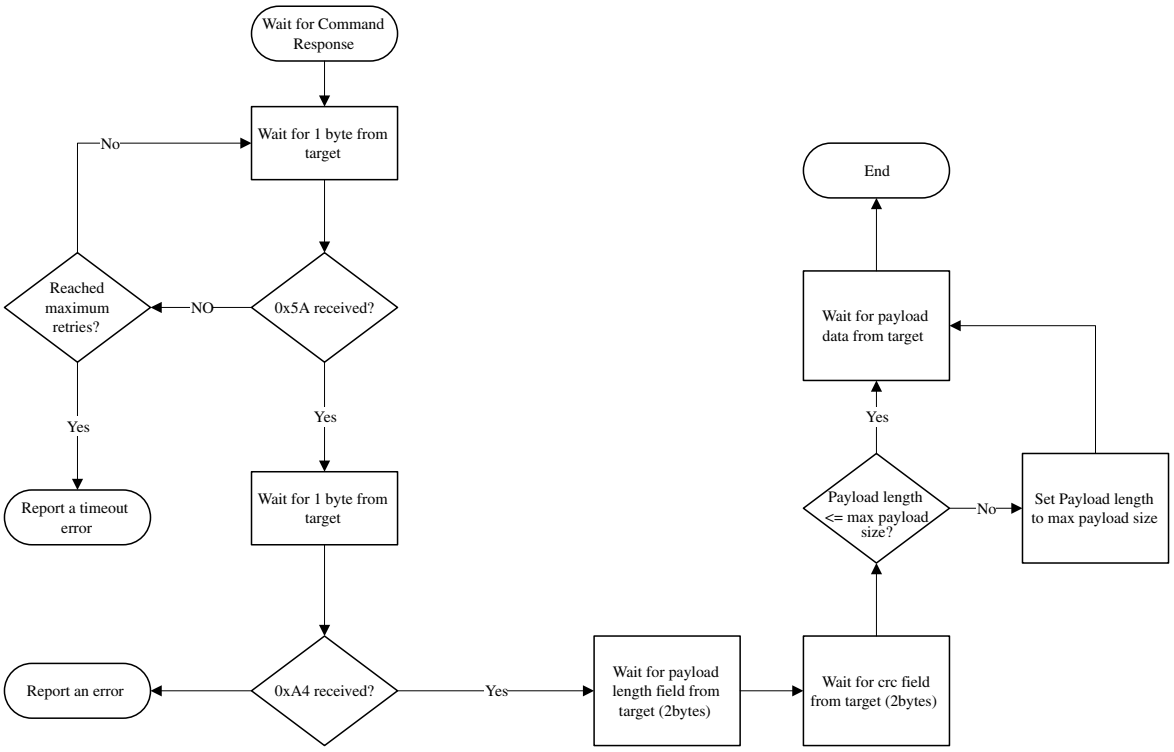


Figure 46. Host reads a command response from target via LPUART

For information on commands and formats, see ISP packet type.

19.3.8 LPI2C ISP

The bootloader supports In-System Programming or serial boot via the LPI2C peripheral, where the LPI2C peripheral serves as the LPI2C slave. The bootloader uses 10h as the 7-bit LPI2C slave address and supports up to 400 kbit/s speed during transfer. The maximum supported LPI2C baud rate depends on the core clock frequency when the bootloader is running. The typical baud rate is 400 kbit/s with factory settings.

The LPI2C peripheral serves as an LPI2C slave device, hence each transfer should be started by the host, and each outgoing packet should be fetched by the host as well.

- An incoming packet is sent by the host with a selected LPI2C slave address and the direction bit is set to write.
- An outgoing packet is read by the host with a selected LPI2C slave address and the direction bit is set as read.
- 00h is sent as the response to host if the target is busy with processing or preparing data.

The following charts show the communication flow of the host reading the ACK packet and the corresponding responses from the target.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

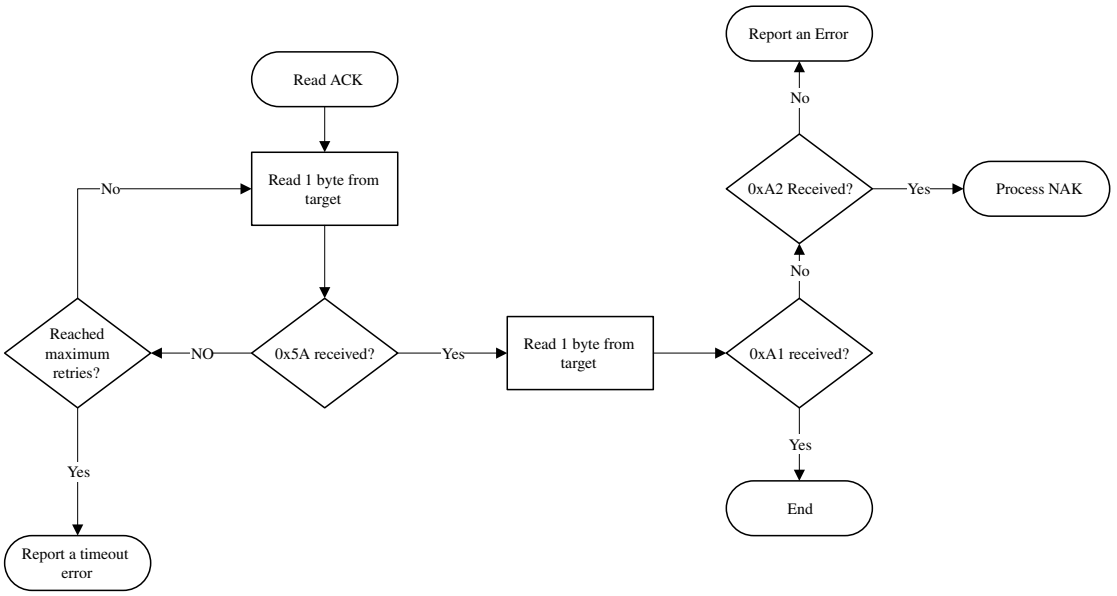


Figure 47. Host reads ACK packet from target via LPI2C

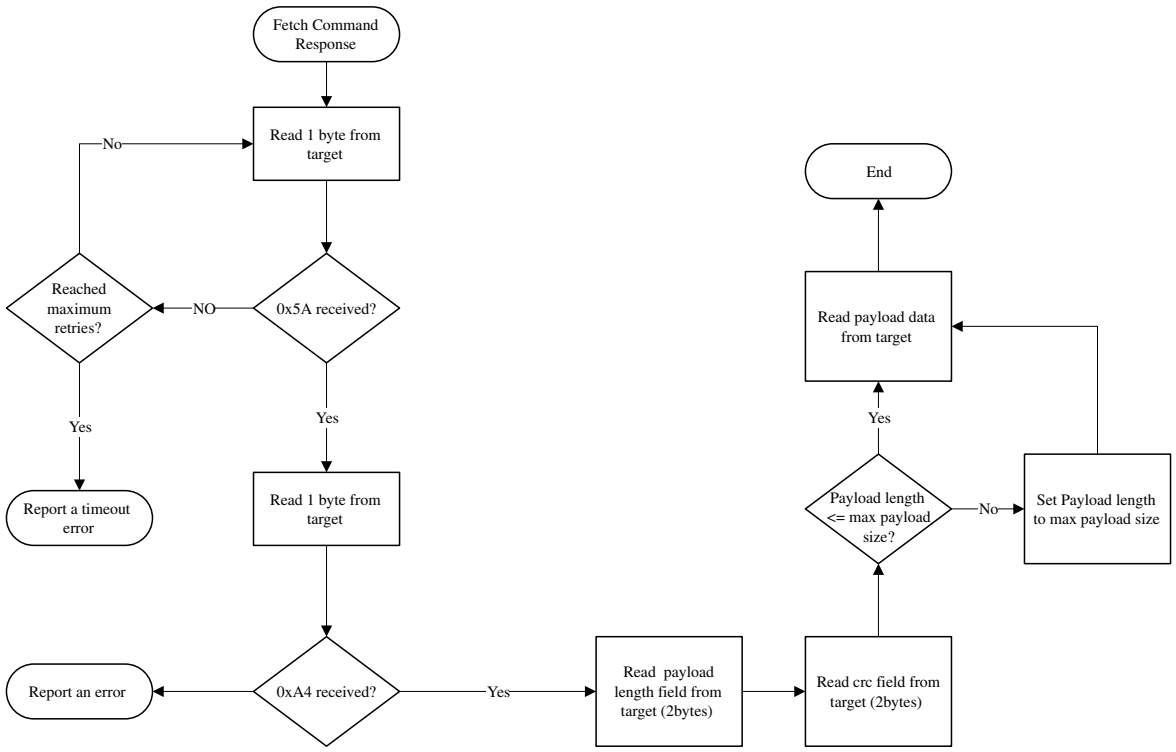


Figure 48. Host reads response from target via LPI2C

For information on commands and formats, see ISP packet type.

19.3.9 LPSPI ISP

The bootloader supports In-System Programming or serial boot via the LPSPI peripheral.

The maximum supported baud rate of the LPSPI depends on the core clock frequency when the bootloader is running. The typical baud rate is 2 Mbit/s with the factory settings. The actual baud rate is up to 12 Mbit/s when the core is running at high-speed boot

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

mode. The LPSPI peripheral in the bootloader serves as an LPSPI slave device. Each transfer, therefore, must be started by the host, and each outgoing packet should be fetched by the host as well.

Compared to LPUART and LPI2C peripherals, the transfer on LPSPI is slightly different:

- The host receives 1 byte after it sends out any byte.
- Received bytes should be ignored when the host is sending out bytes to the target
- The host starts reading bytes by sending 00hs to target
- The byte 00h is sent as a response to host if the target is under the following conditions:
  - Processing incoming packet
  - Preparing outgoing data
  - Received invalid data

The following flowcharts show how the host reads a ping response, an ACK and a command response from target via LPSPI without the nIRQ pin enabled.

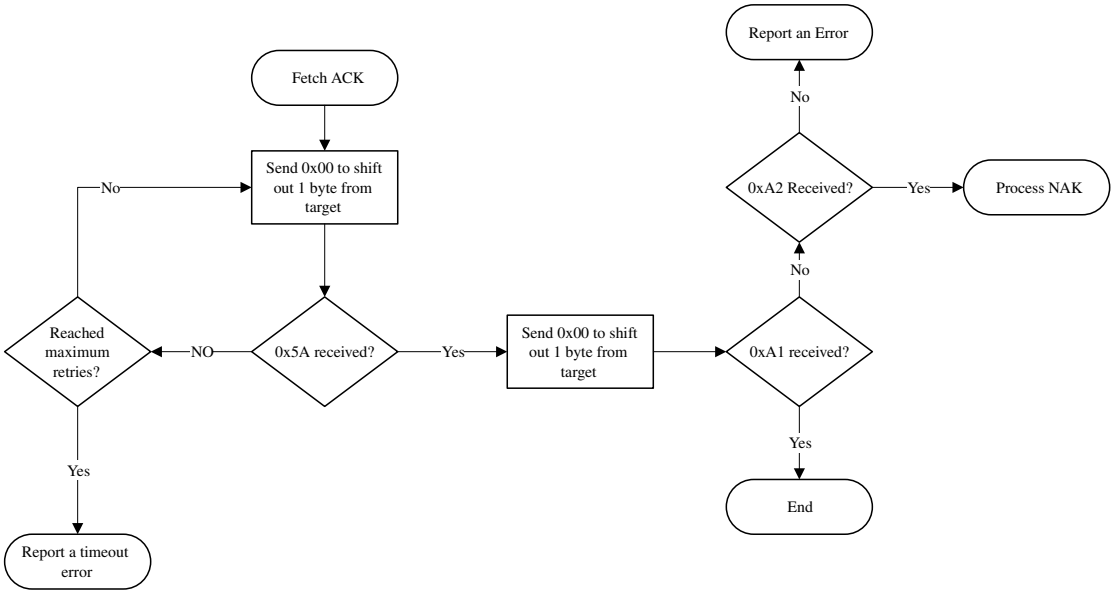


Figure 49. Host reads ACK from target via LPSPI

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

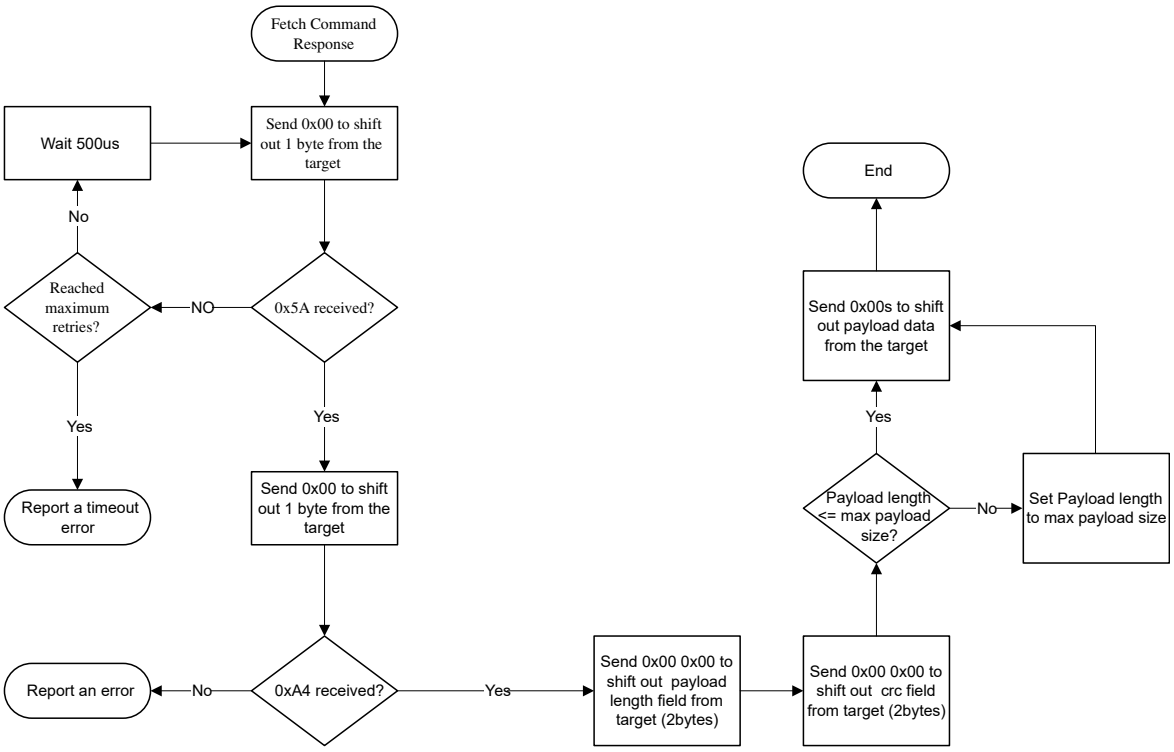


Figure 50. Host reads response from target via LPSPI

For information on commands and formats, see ISP packet type.

19.4 Debug Mailbox

19.4.1 Overview

The debug mailbox (DBGMB) communication protocol is implemented in bootloader based on DBGMB module.

19.4.2 Quick guide of using Debug Mailbox (DBGMB)

To start the communication with the bootloader via DBGMB, the debugger needs to select DM-AP and set the RESYNCH_REQ bit and the CHIP_RESET_REQ bit to 1 in the CSW register. Following a successful initial re-synchronization and reset, communication by the debugger to the device is achieved using 32-bit DM-AP command writes to the REQUEST register in the DBGMB (DM-AP commands are shown in table below). The debugger can read results of communications via the RETURN register. Response codes are shown in DM-AP return codes.

The example below shows how to enter debug mailbox and do the communication:

```
// Pseudo Code Syntax// ----------------------------------------------------------------------
                // WriteDP <register> <value>// value = ReadDP <register>
                // AP transactions presume the DM AP is selected
                // WriteAP <register> <value>// value = ReadAP <register> <value>
                // ----------------------------------------------------------------------
                // Read AP ID register to identify DM AP at index 1
                // Select DM AP index 1
                WriteDP 1 0x01000000
                // Write DM RESYNC_REQ + CHIP_RESET_REQ
                WriteAP 0 0x21
                readAP 0
                // Write DM-AP Command START DM-AP (1) to REQUEST register (1)
```


All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

```
                WriteAP 1 0x1
                readAP 2
```

19.4.3 DBGMB commands

This section describes the request and response message formats and available mailbox commands.

19.4.3.1 Request packet layout

The first word transmitted in a request is a header word containing the command ID and the number of following data words. The command packet is sent to the device by writing 32 bits at a time to the REQUEST register. When sending command packets greater than 32 bits, the debugger must read an ACK_TOKEN in the RETURN register before writing the next 32 bits. The 32-bit words quantified by the header follow the header itself.

Table 132. Request register byte description

Word	Byte 0	Byte 1	Byte 2	Byte 3
0	commandID[7:0]	commandID[15:8]	dataWordCount[7:0]	dataWordCount[15:8]
1	data	—	—	—

The C structure definition for a request is:

```
struct dm_request {
        uint16_t commandID;
        uint16_t dataWordCount;
        uint32_t data[ ];
        };
```

19.4.3.2 Response packet layout

The first word transmitted in a response is a header word containing the command status and the number of following data words. The command response can be read 32 bits at a time through Return Value (RETURN). The initial 32 bits contains the response header, as shown in the table below. When reading a response longer than 32 bits, the debugger writes the ACK_TOKEN to REQUEST after every read until the full response packet is received.

Table 133. Response register byte description

Word	Byte 0	Byte 1	Byte 2	Byte 3
0	bits[7:0]:commandStatus[7:0]	bits[7:0]:commandStatus[15:8]	bits[7:0]:dataWordCount[7:0]	bits[6:0]:dataWordCount[14:8] bits[7]:new_protocol[15]
1	data	—	—	—

The C structure definition for a response is:

```
struct dm_response {
        uint16_t commandStatus;
        uint16_t dataWordCount;
        uint32_t data[];
        };
```


All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

19.4.3.3 DM-AP commands

DM-AP commands are written to the REQUEST register. An Exit DM_AP command follows one or more of the DM-AP commands in the table below to resume the normal device boot flow. This sequence does not occur after the Enter ISP Mode and Start Debug Session commands.

Table 134. DM-AP commands

Name	Command code	Parameter and response	Description
Start DM-AP	01h	Parameters: None Response: 32-bit status (returns ACEDh)	Causes the device to enter DM-AP command mode.
GET CRP Level	02h	Parameters: None Response: 32-bit value	Returns the value of current lifecycle.
Mass Erase	03h	Parameters: None (lifecycle is Develop) 4-word passwords (lifecycle is In-field1/2) Response: 32-bit status	Erase the PFlash regions (offsets 0x0000–0xFFFF and 0x1C000–0x1FFFF) and IFR0 sectors 1 through 3.
Exit DM_AP	04h	Parameters: None Response: 32-bit status	Causes the device to exit DM-AP command mode. The device returns to normal boot path.
Start debug Session	07h	Parameters: None Response: 32-bit status	Bootloader boots a dummy image, while (1) loop in IFR0.
Advance Lifecycle	17h	Parameters: 1-word lifecycle value Response: 32-bit status	Advance lifecycle from Develop to other lifecycles.

19.4.4 DM-AP return codes

Return codes for DM-AP commands are listed below. These commands can be read from the RETURN register.

Table 135. DM-AP return codes

Return code	Descriptions
0000_0000h	Command succeeded
0010_0002h	Command not recognized. A command was received other than the ones defined above.
0010_0003h	Command failed

19.4.5 DBGMB protocol

Bootloader implements debug mailbox protocol to interact with host debug systems over the SWD interface. The protocol has following features:

- Request/response based
- Support for relatively large command and response data

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

- All commands and responses are 32-bit word aligned
- Supports data above 32-bits by using an ACK_TOKEN that moderates the transfer in 32-bit value chunks.
- Requests and responses use the same basic structure.

19.4.6 Password-based debug authentication (Only at In-field1)

The bootloader supports password-based debug authentication. When the device is in the in-field1 lifecycle state, the debug port can be opened after successful debug authentication.

Steps for password-based debug authentication:

1. Set the password.

   During the Develop lifecycle, program a valid password (not 00000000h or FFFFFFFFh) into the CMPA DBG_AUTH_PASSWORD field.

2. Authenticate in In-field1 lifecycle.
   - Enter DBGMB.
   - Write the password into DBGMB registers 80h − 8ch.
   - If the entered password matches the stored value in DBG_AUTH_PASSWORD, authentication succeeds.

Post-authentication options:

- **Call Exit DM command**

  Allows the bootloader to continue the normal boot flow. If the boot image is valid and boots successfully, the debug port becomes accessible.

- **Call Start Debug Session command**

  Used when no valid image exists or the image is corrupted. Boots a dummy image (while(1) loop in IFR0) to enable debugging activities.

The example below shows how to perform password-based debug authentication after setting the password in the CMPA password region.

```
//Active DebugMailBox
                writeDP 2 0x01000000
                readDP 0
                writeAP 0 0x21
                readAP 0
                // send ENTER_DEBUGGER_MAILBOX command
                writeAP 1 0x01
                readAP 2 // will return 0xACED
                // start debug auth write password to registers
                writeDP 2 0x01000080
                writeAP 0 0x7c8c9cac //pasword0
                writeAP 1 0x1b3b5b7b //pasword1
                writeAP 2 0xa5a55a5a //pasword2
                writeAP 3 0xeccebeeb //pasword3
                //start debug session
                writeDP 2 0x01000000
                writeAP 1 0x7
                readAP 2 // will return 0x8000_0000 and enters while(1) loop
```

19.4.7 Debug session usage instructions

The method used to initiate a debug session depends on the device state and the intended debugging scenario. The following scenarios are covered in the subsequent sections:

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

- Debug session with uninitialized or invalid image

  If no valid boot image is present, or image is corrupted, debugger can't attach to device through standard debug connection flow.

- Debug session with bootloader ISP mode entered

  Program control (PC) resides in ISP mode which is entered when the ISP pin is asserted during device reset. In this mode, the debug interface is disabled.

- Debug session attaching to a running target

  In this scenario, the bootloader has successfully launched the application image. The host connects to the device without reprogramming or resetting it (commonly referred to as a debug attach).

NOTE

The procedures described in the following sections apply only to the Develop lifecycle state. In the In-field1 lifecycle, debug authentication must be successfully completed before initiating a debug session. Debug session attachment is not supported in other in-field lifecycle states.

19.4.7.1 Debug session with uninitialized or invalid image

When the device boots without a valid image in the boot media, program control (PC) remains in the bootloader. For security reasons, debug access is disabled during bootloader execution. If the image is corrupted, the device may repeatedly reset and return to the bootloader, creating a continuous boot–reset cycle. This behavior prevents the debugger from attaching to the target. This section describes methods for establishing a debug session under these conditions.

To ensure that the state machine controlling debug mailbox commands is in a known state, the debugger can reset this logic via the following steps:

1. Write 1 to CSW[RESYNCH_REQ] to request resynchronization.
2. Write 1 to CSW[CHIP_RESET_REQ] to reset the chip.
3. Read CSW.
4. Wait until the chip has completed resynchronization, identified by reading a value of 0 from the lower 16 bits of CSW.

To start a debug session and control the exchange of debug information, the debugger uses the DM-AP commands:

1. Following a successful initial resynchronization, communicate with the chip via 32-bit DM-AP commands to the REQUEST register in the debug mailbox.
2. Read results via the RETURN register. To ensure that the transactions have completed successfully, the debugger must poll the RETURN register as it polled the CSW register following a resynchronization request.
3. View the results in Response packet.

To initiate a debug session over SWD while debug is disabled:

1. The debug system must issue a Start Debug Session command to the Bootloader-based boot code, following the Debug session protocol.
2. Upon receiving the command, the boot code disables any unwanted peripheral and manages NXP secrets before enabling debug access.
3. After enabling debug access, the Bootloader enters a while(1) loop.

Once the Start Debug Session command and chip reset have executed successfully, the AP for CPU0 is accessible. You can use it to set breakpoints and perform other tasks, as with other Arm Cortex-M devices. Once a debug connection is successfully established, you can set the required breakpoints in the code.

The code sample below shows how to initiate a debug session for the scenarios described above:

```
// WriteDP <register> <value>// value = ReadDP <register>
// AP transactions presume the DM AP is selected
```


All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

```
// WriteAP <register> <value>// value = ReadAP <register> <value>
// -------------------------------------------------------------------------
// Read AP ID register to identify DM AP at index 1
// Select DM AP index 1
WriteDP 1 0x01000000
// Write DM RESYNC_REQ + CHIP_RESET_REQ
WriteAP 0 0x21
readAP 0
// Write DM-AP Command Start debug Session to REQUEST register (1) under Develop lifecycle
WriteAP 1 0x7
readAP 2
```

19.4.7.2 Debug session with Bootloader ISP mode entered

In this scenario, the ISP pin is asserted at reset, causing the bootloader to enter ISP mode. The debug interface is disabled during bootloader execution. To establish a debug session in this condition, see Debug session with uninitialized or invalid image.

19.4.7.3 Debug session attaching to a running target

In this scenario, the device has completed boot and is executing an application that has not disabled debug access. The host system can establish a debug connection without issuing a reset or modifying the application. To halt execution at the end of the bootloader, the debugger can configure a data watchpoint on a read access to address 0x400E861C. This causes the core to stop immediately upon reaching this point. Under these conditions, the CPU0 Access Port (AP) remains accessible, allowing breakpoints to be set without requiring re-synchronization of the mailbox hardware.

19.5 Run bootloader API

The bootloader provides an API for the user application to enter the ISP mode based on the designated ISP interface mode.

Prototype

```
void (*runBootloader)(void *arg);
```

Table 136. runBootloader API arg fields

Field	Offset	Description
Tag	[31:24]	Fixed value: 0xEB (Enter Boot)
Boot mode	[23:20]	0 - Reserved 1 - Enter ISP mode
Boot interface	[19:16]	ISP boot mode: 0h - UART 1h - I2C 2h - SPI 7h - Auto detection Other values - Reserved
Reserved	[15:0]	

In the application image, write the following code or call runBootloader API.

```
typedef struct
{
```


All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

```
    __I void (*runBootloader)(void *arg);           //!< Function to start the bootloader executing.

} BL_API_TYPE;

#define BL_API_BASE (0x1000f00u)
#define BL_API ((BL_API_TYPE *)BL_API_BASE)

uint32_t tag = 0xEB110000;
BL_API->runBootloader(&tag);
```

After executing the code above, the bootloader starts execution and enters I2C ISP.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Chapter 20
Clocking

20.1 Introduction

The main component of the clocking architecture is the System Clock Generator (SCG) module. The SCG controls multiple clock sources that are distributed to the main CPU platform, the memory modules and the peripheral modules. The peripheral modules have an *interface* clock that is used by the CPU/DMA to interface with each module's registers. When applicable, modules may also have *functional* clocks to provide the main timing function of the module's application. For example, functional clocks can be used to source the baud rate for a serial communications peripheral or to clock the counter of a timer peripheral.

The SCG controls the on-chip clock sources, including FRO144M and FRO12M. It also generates the clock sources for peripherals functional clock. A clock mux is implemented in SCG-Lite to select the clock source for main_clk.

The MAIN_CLK can be sourced directly from the following:

- FRO144M
- FRO12M
- OSC32K

20.2 Configure main clock and system clock

The clock source for the registers and memories is derived from main clock. The main clock can be selected from the sources listed in step 1 below.

The CPU_CLK derived from MAIN_CLK, also called the system clock, clocks the core, the memories, and the peripherals (register interfaces and peripheral clocks).

1. Select the main clock in the SCG_RCCR register. The following options are available:
   - 12 MHz output (FRO12M) from internal oscillator.
   - 32 kHz OSC output (OSC32K).
   - FRO high speed output (fro_hf) from FRO144M, fro_hf_gated is the gated version of fro_hf. By default, they are 48 MHz. fro_hf is the default main clock.
2. Select the divider value for the system clock AHBCLKDIV register in the System Controller module.
3. Enable the clock for the memories and peripherals used in the application.
4. Peripheral clock must be disabled registers in SYSCON module when performing software reset on the peripherals. Once reset assertion and de-assertion are completed, the respective peripheral clock can be enabled.

20.3 Clock generation

The system control block facilitates the clock generation. Many clocking variations are possible. Module clocking gives an overview of potential clock options. Table 137 describes signals on the clocking diagram. The maximum clock frequency is 144 MHz.

NOTE

The indicated clock multiplexers shown in Module clocking are synchronized. In order to operate, the currently selected clock must be running, and the clock to be switched to must also be running so the multiplexer can gracefully switch between the two clocks without glitches. Other clock multiplexers are not synchronized. The output divider can be stopped and restarted gracefully during switching if a glitch-free output is needed.

The FRO12M provides a 1 MHz clock (clk_1M). It is used as Watchdog 0 independent clock source. The accuracy of this clock is limited to a couple percent over temperature (see Data sheet for specification), voltage, and silicon processing variations after

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

trimming made during assembly. To determine the actual Watchdog oscillator output, use the frequency measure block. See Chip-specific FREQME information.

NOTE

The maximum allowed frequency for the main clock and system clock (to CPU0, AHB bus, Sync, etc.) is 144 MHz.

Table 137. Clocking diagram signal name descriptions

Name[1]	Description
OSC32K[0]	It is the 32.768KHz clock output from OSC32K. It is the clock of peripherals in System domain.
OSC32K[1]	It is the 32.768KHz clock output from OSC32K. It is the clock of peripherals in CORE domain.
clk_16k[0]	It is the 16.384 kHz clock output from FRO16K. It is the clock of peripherals in system domain.
clk_16k[1]	It is the 16.384 kHz clock output from FRO16K. It is the clock of peripheral in core domain.
clk_in	It is the clock that comes from the external oscillator. It is the clock from CLKIN pin.
fro_hf/fro_hf_gated	It is the clock output from FRO144M. The frequency is controlled by SCG-Lite SCG_FIRCCFG[FREQ_SEL] register.
fro_hf_div	It is the clock divided by fro_hf_gated.
clk_48M	It is the clock derived from from FRO144M. SCG FIRCCFG.FREQ_SEL[1] should be 0, and clk_48M is 48 MHz.
fro_12M	It is the 12MHz clock output from FRO12M.
fro_12m_div	It is clock derived from FRO12M.
clk_1M	It is the 1 MHz clock output from FRO12M.
main_clk	It is main clock used by the CPU, AHB bus, APB bus, IPS bus, and some peripherals. The main clock and its source selection are shown in Figure 53.
CPU_CLK	It is the clock of CPU.
SYSTEM_CLK	It is the clock of AHB bus, APB bus, IPS bus. The frequency is same with CPU_CLK.
SLOW_CLK	It is the SYSTEM_CLK divided by 4.

1. See the clock frequency table for max frequencies of each of these clocks.

Table 138. Clock frequency table

Clock Name	Max. Clock Frequency
	OD Mode
OSC32K[0]/OSC32K[1]	32.768 kHz
clk_16k[0]/ clk_16k[1]	16.384 kHz
clk_in	50 MHz (max)

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table 138. Clock frequency table (continued)

Clock Name	Max. Clock Frequency
	OD Mode
fro_hf/fro_hf_gated	144 MHz
fro_hf_div	72 MHz
clk_48M	48 MHz
fro_12M	24 MHz
fro_12m_div	24 MHz
clk_1M	1 MHz
main_clk	144 MHz
CPU_CLK	72 MHz
SYSTEM_CLK	72 MHz
SLOW_CLK	18 MHz

20.3.1 Peripheral bus clock

The following peripherals use SLOW_CLK as APB/IPS clock. All other peripherals use SYSTEM_CLK as APB/IPS bus clock. The AHB interface peripherals should be SYSTEM_CLK.

- VBAT/OPAMP0
- LPTMR0
- CMP0
- PORT0~3
- ATX0
- CMC
- SCG-Lite
- SPC
- WUU
- FMU
- RTC
- TRNG

20.3.2 Peripheral clock summary

Modules	Bus Clock Source and Frequency	Max Functional Clock	Max Functional IP Clock
CPU and System Modules			
CM23	SD: 72 MHz	—	—
CPU_CLK/SYSTEM_CLK	SD: 72 MHz	—	—

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Modules	Bus Clock Source and Frequency	Max Functional Clock	Max Functional IP Clock
SLOW_CLK	SD: 1/4 SYSTEM_CLK	—	—
DMA0	SYSTEM_CLK	—	—
CRC	SYSTEM_CLK	—	—
AHB Matrix	SYSTEM_CLK	—	—
APB Bridge	SYSTEM_CLK	—	—
AIPS Bridge	SYSTEM_CLK	—	—
All System Masters' AHB Bus	SYSTEM_CLK	—	—
SYSTICK CLK		SD: 24 MHz	—
SYSCON	SYSTEM_CLK	—	—
SCG	SLOW_CLK	—	—
SPC	SLOW_CLK	—	—
WWDT0	SYSTEM_CLK	SD: 1 MHz	—
WWDT1	SYSTEM_CLK	SD: 24 MHz	—
CMC	SLOW_CLK	—	—
RMC	SLOW_CLK	—	—
Peripheral Input Mux	SYSTEM_CLK	—	—
TRNG	SLOW_CLK	FRO12M	—
Debug Modules			
JTAG/SWD_CLK	—	—	SD: 24 MHz
TRACKCLKIN	—	—	SD: 24 MHz
Memory Modules			
SRAM	SYSTEM_CLK	—	—
EIM	SYSTEM_CLK	—	—
ERM	SYSTEM_CLK	—	—
FMC Controller	SYSTEM_CLK	—	—
FMU APB Interface	SLOW_CLK	—	—
FMU AHB Interface	SYSTEM_CLK	—	—
Communication Modules			
LPUART0~4	SYSTEM_CLK	SD: 144 MHz	SD: 24 Mbps
LPSPI0	SYSTEM_CLK	SD: 144 MHz	(Reference Package and IO Chapter)
LPI2C0	SYSTEM_CLK	SD: 72 MHz	3.4 Mbps

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Modules	Bus Clock Source and Frequency	Max Functional Clock	Max Functional IP Clock
Timer Modules			
LPTMR0	SLOW_CLK	SD: 24 MHz	—
CTIMER0~1	SYSTEM_CLK	SD: 144 MHz	—
FREQME	SYSTEM_CLK	SD: 72 MHz	—
RTC	SLOW_CLK	32.768 kHz	—
FlexPWM	SYSTEM_CLK	—	—
HMI Modules			
GPIO0~3	SYSTEM_CLK	—	—
PORT0~3	SLOW_CLK	—	—
Analog Modules			
ADC	SYSTEM_CLK	SD: 48MHz	—
CMP0	SLOW_CLK	24 MHz in all mode	—
OPAMP	SLOW_CLK	—	—
POR/VMON	SLOW_CLK	—	—
PMC	SLOW_CLK	10 MHz	—
MISC Modules			
VBAT Wrapper	SLOW_CLK	32.768 kHz	—
WUU	SLOW_CLK	16.384 kHz	—
CLKOUT	—	SD: 72 MHz	—

20.3.3 Module clocking

The following figures show the clock generation diagrams of the device's modules.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

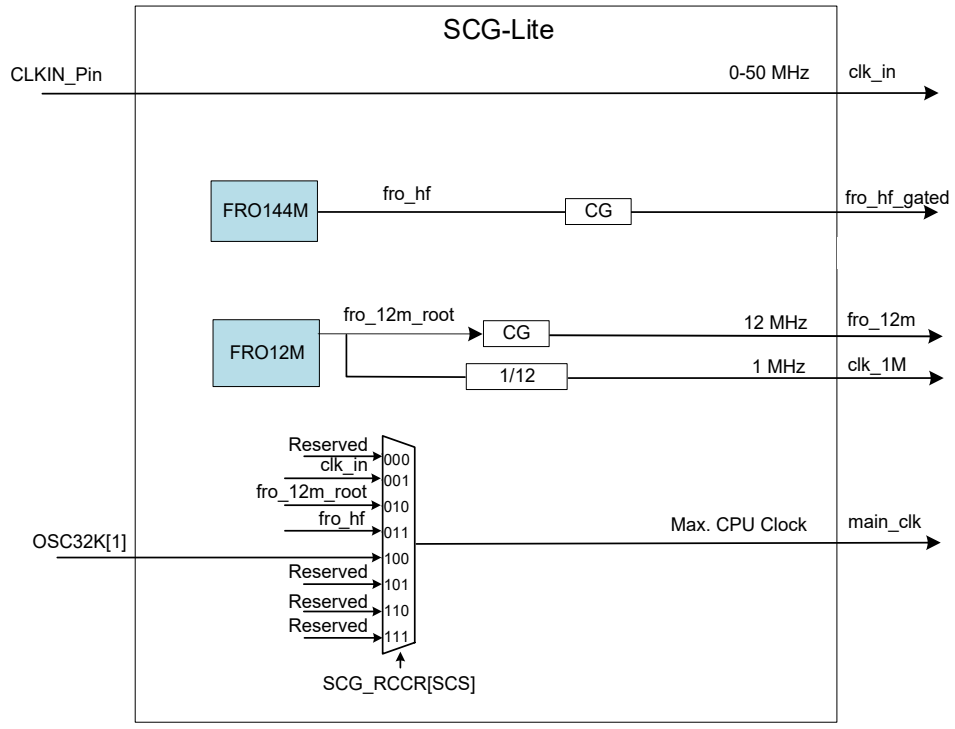


Figure 51. SCG clock

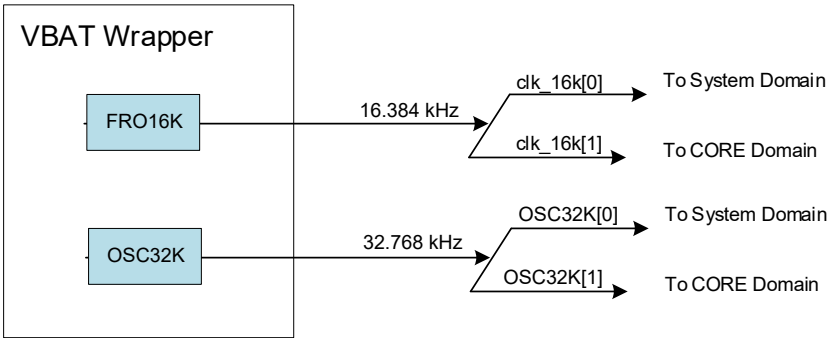


Figure 52. VBAT clock

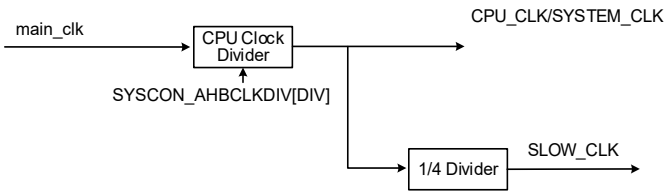


Figure 53. Main CPU clock

© 2026 NXP B.V. All rights reserved.

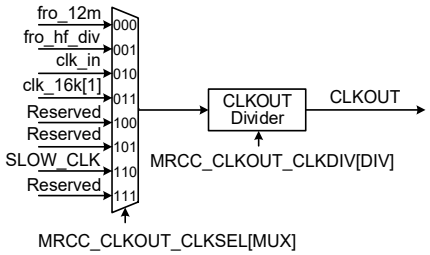


Figure 54. CLKOUT clock

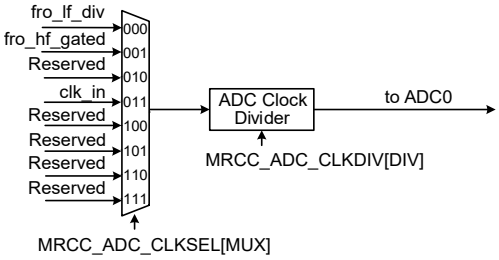


Figure 55. ADC0 clock

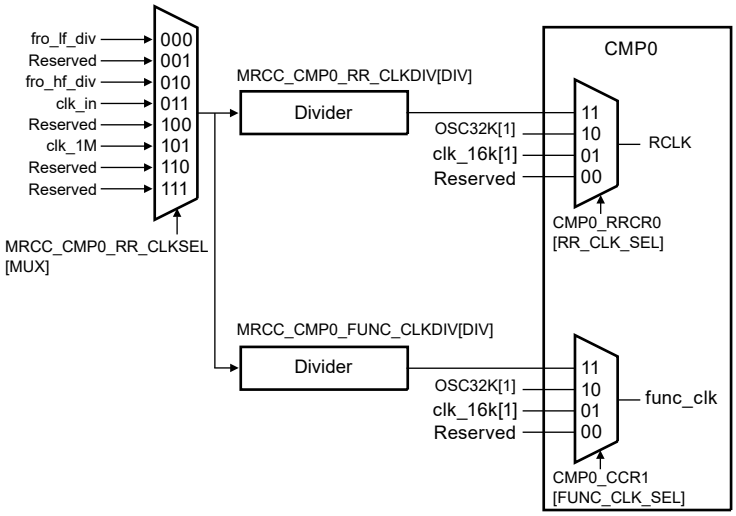


Figure 56. LPCMP0 clock

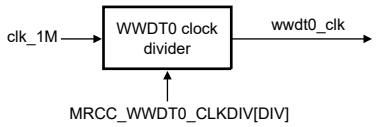


Figure 57. WWDT0 clock

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

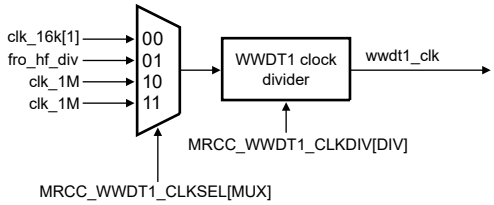


Figure 58. WWDT1 clock

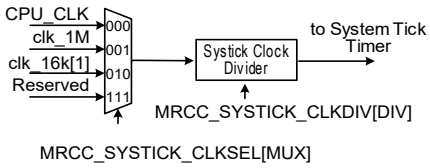


Figure 59. Systick clock

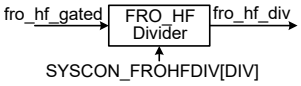


Figure 60. FRO HF clock

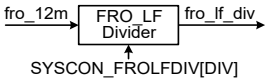


Figure 61. FRO LF clock



Figure 62. FlexPWM clock

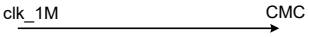


Figure 63. CMC clock

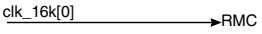


Figure 64. RMC clock

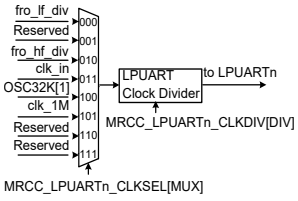


Figure 65. LPUART clock

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

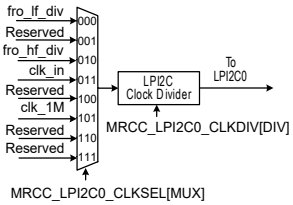


Figure 66. LPI2C clock

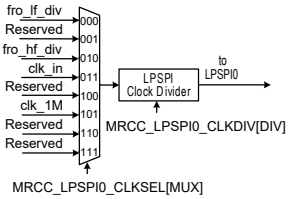


Figure 67. LPSPI clock

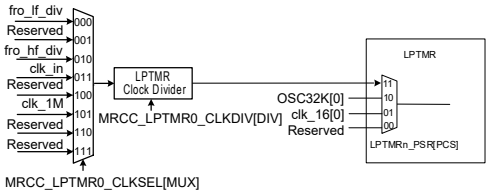


Figure 68. LPTMR clock

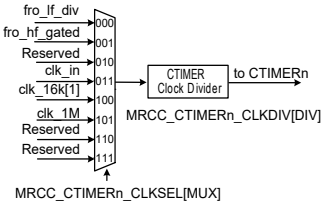


Figure 69. CTIMER clock

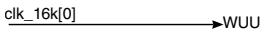


Figure 70. WUU clock

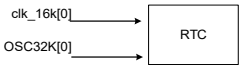


Figure 71. RTC clock

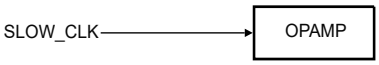


Figure 72. OPAMP clock

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

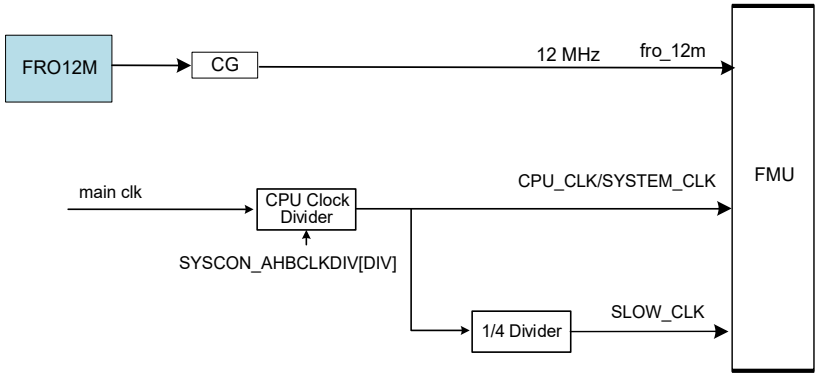


Figure 73. FMU clock

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Chapter 21
System Clock Generator (SCG)

21.1 Chip-specific SCG information

Table 139. Reference links to related information

Topic	Related module	Reference
Full description	SCG	SCG
System memory map		Memory map
Clocking		Clock distribution
Power Management		Power Management
Signal multiplexing	Port control	Signal multiplexing

21.1.1 Module instances

This chip contains one instance of the SCG module, SCG0.

21.2 Overview

The System Clock Generator (SCG) module provides the main clock and other peripheral clocks for the MCU. The SCG takes clock inputs from a variety of sources and generates the clocks that the MCU requires.

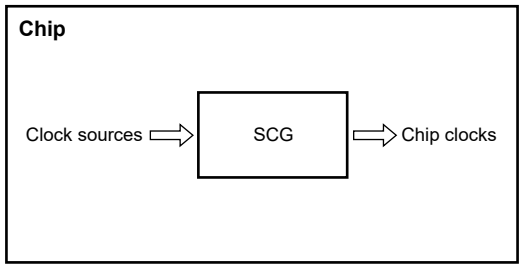


Figure 74. System Clock Generator (SCG) in MCU

The clock source inputs available for the SCG are as follows:

- System oscillator clock (SOSC)

NOTE

SOSC can only be used under bypass mode (SOSCCFG[EREFS] = 0).

- Slow internal reference clock (SIRC)
- Fast internal reference clock (FIRC)
- Real-Time oscillator clock (ROSC)

© 2026 NXP B.V. All rights reserved.

21.2.1 Block diagram

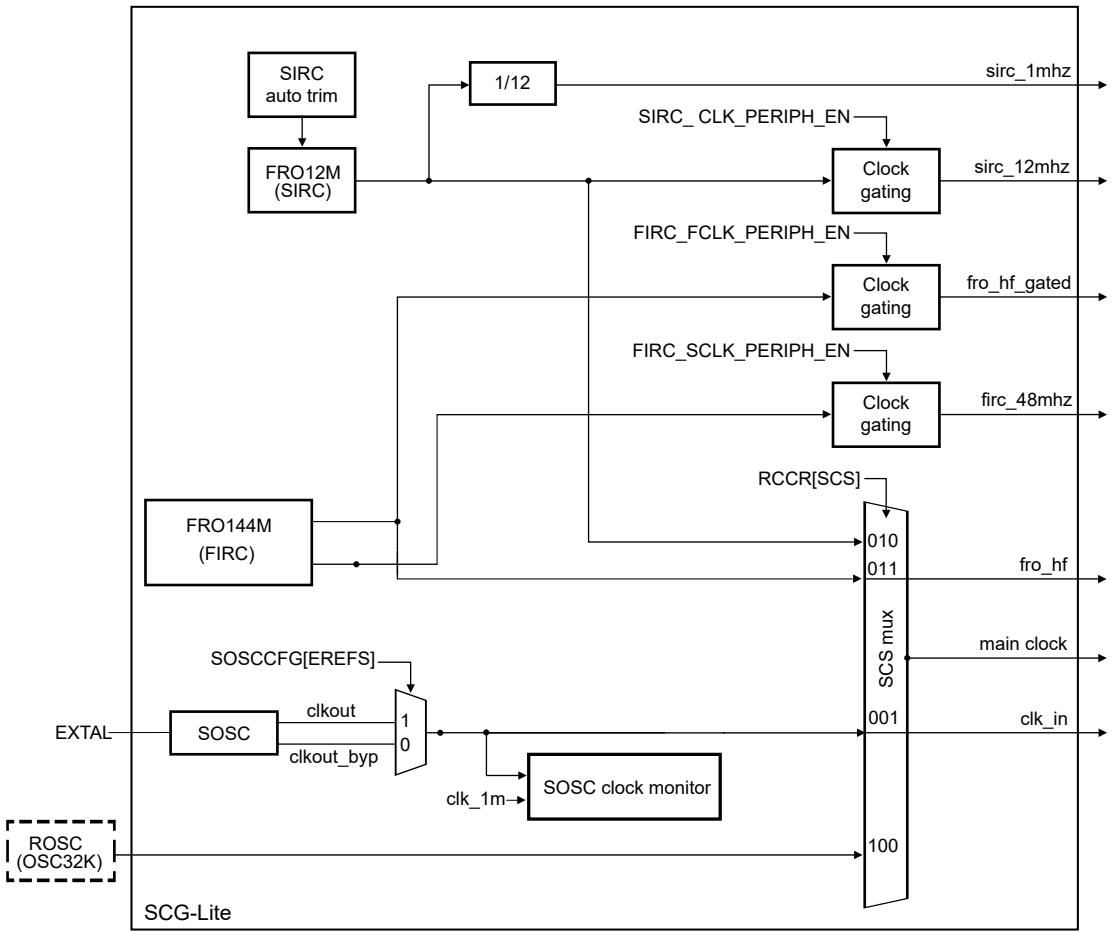


Figure 75. Block diagram

21.2.2 Features

Key features of the SCG module are:

- 2 Internal Reference Clock (IRC) generators
  - FIRC with trim capability
  - FIRC can output 144/72/48/36 MHz clock by register configuration
  - FIRC can limit output frequency by frequency phantom
  - FIRC can output 48 MHz clock continuously
  - SIRC with trim capability
  - SIRC can output 12 MHz clock and 1 MHz clock divided from it
  - Either of the IRC generators can be selected as the clock source for the system clocks
- Real Time Oscillator Clock (ROSC)
  - Can be selected as the clock source for the device. See the Clocking chapter to ascertain which signal the ROSC input maps to.
- Clock monitor with reset and interrupt request capability for SOSC, ROSC clocks

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

21.3 Functional description

21.3.1 SCG clock mode transitions

The system boots up from the FIRC source. The system clock can be switched among the following clock sources:

- FIRC
- SIRC
- SOSC

NOTE

SOSC can only be used under bypass mode (SOSCCFG[EREFS] = 0).

- ROSC

The SCG restricts programming into invalid clock modes, and ignores writes to the System Clock Source (SCS) bits. When a transition between run modes or a transition from run mode into wait mode occurs, the SCG completes the switch to the corresponding clock mode as defined in the Run Clock Control Register (RCCR). After that, the system switches to the selected run/wait mode.

The following table describes the modes of operation of the SCG.

Table 140. SCG modes of operation

Mode	Description
SOSC	The SOSC mode is entered when the following occur: • the device is in run mode and 001 is written to RCCR[SCS] • SOSCCSR[SOSCVLD] = 1b In SOSC bypass mode, the system clocks are from EXTAL clock.
SIRC	The SIRC mode is entered when the following occur: • the device is in run mode and 010 is written to RCCR[SCS] • SIRCCSR[SIRCVLD] = 1b In SIRC mode, the system clocks are derived from SIRC.
FIRC	The FIRC mode is the default clock mode of operation or it is entered when the following occur: • the device is in run mode and 011 is written to RCCR[SCS] • FIRCCSR[FIRCEN] = 1b • FIRCCSR[FIRCVLD] = 1b In FIRC mode, the system clocks are derived from FIRC. Eight frequency settings are available for FIRC clock as described in FIRCCFG[FREQ_SEL]. The changes to FIRC frequency settings are ignored when the FIRC clock is enabled.
ROSC	The ROSC mode is entered when the following occur: • the device is in run mode and 100 is written to RCCR[SCS] • ROSCCSR[ROSCVLD] = 1b In ROSC mode, the systems clocks are derived from the external ROSC.

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table 140. SCG modes of operation (continued)

Mode	Description
Stop	The Stop mode is entered whenever the MCU enters a Stop state. Power modes are chip-specific. For power mode assignments, see Power Management chapter that describes module operation in low-power modes. Entering Stop mode, all SCG clock signals are static, including SCG system clocks. There are some exceptions. The following clocks can continue to run and stay enabled when all the specific conditions are true. SIRC is available in Deep Sleep mode when all the following conditions are true: • SIRCCSR[SIRCSTEN] = 1b FIRC is available in Deep Sleep mode when all the following conditions are true: • FIRCCSR[FIRCEN] = 1b • FIRCCSR[FIRCSTEN] = 1b

21.3.2 SCG SOSC operation

The following table lists the SOSC configurations and SOSC output clock.

NOTE

SOSC can only be used under bypass mode (SOSCCFG[EREFS] = 0).

21.3.3 SCG LOC operation

The SCG supports the loss of clock (LOC) monitoring for the external reference clocks including SOSC and ROSC reference clocks. To enable the LOC monitor, the CME bit for each of these clocks must be programmed to 1. To monitor these external reference clocks, the SCG uses an internal 31.25 KHz reference clock derived by dividing the SIRC internal clock. So prior to enabling the LOC for either of these external reference clocks, it is required that the SIRC is programmed to be enabled.

The LOC monitors, when enabled, generate a system reset or system interrupt request depending on the settings of the CMRE (see the CMRE description for each of the clocks for further information). The frequency conditions for the external reference clocks that trigger a LOC event are listed below:

NOTE

F_{int} = SIRC divided down to 31.25 kHz

- ROSC loss of external clock minimum frequency
  - When ROSC clock frequency is above (3/5) F_{int}, the chip is never reset or generate an interrupt.
  - When ROSC clock frequency is between (2/5) F_{int} and (3/5) F_{int}, the chip might reset or generate an interrupt (depending on the phase dependency between ROSC clock and reference clock)
  - When ROSC clock frequency is below (2/5) F_{int}, the chip always resets or generates an interrupt.
- SOSC loss of external clock minimum frequency
  - When SOSC clock frequency is above (16/5) F_{int}, the chip is never reset or generate an interrupt.
  - When SOSC clock frequency is between (15/5) F_{int} and (16/5) F_{int}, the chip might reset or generate an interrupt (depending on the phase dependency between SOSC clock and reference clock)
  - When SOSC clock frequency is below (15/5) F_{int}, the chip always resets or generates an interrupt.

21.3.4 Clocking

The SCG module has the following input clocks:

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

- Bus clock for register access.
- One of the following as the system clock source:
  - ROSC

The SCG module has the following output clocks:

- Main clock
- FIRC clock (144/72/48/36 MHz)
- SIRC clock (12 MHz and 1 MHz divided from it)
- SOSC clock

NOTE

SOSC can only be used under bypass mode (SOSCCFG[EREFS] = 0).

21.3.5 Resets

These are the SCG resets:

1. The global reset is fed into this module. The entire module is reset by the global reset.
2. The clock monitor generates a reset in case of the following:
   - SOSCCSR[SOSCCM] = 1 & SOSCCSR[SOSCCMRE] = 1 & SOSCCSR[SOSCERR] = 1
   - ROSCCSR[ROSCCM] = 1 & ROSCCSR[ROSCCMRE] = 1 & ROSCCSR[ROSCERR] = 1

On the assertion of any reset source, the FIRC 48 MHz clock is enabled or starts up if not running.

21.3.6 Interrupts

These are the SCG interrupts:

- Clock monitor generates interrupt when any of below happens:
  - SOSCCSR[SOSCCM] = 1 & SOSCCSR[SOSCCMRE] = 0 & SOSCCSR[SOSCERR] = 1
  - ROSCCSR[ROSCCM] = 1 & ROSCCSR[ROSCCMRE] = 0 & ROSCCSR[ROSCERR] = 1
- Interrupt can also be generated when any of below happens:
  - SOSC clock source is not valid. SOSCCSR[SOSCVLD_IE] = 1 & SOSCCSR[SOSCVLD] = 1
  - FIRC clock source is not accurate. FIRCCSR[FIRCACC_IE] = 1 & FIRCCSR[FIRCACC] = 1
  - FIRC trimming error detected. FIRCCSR[FIRCERR_IE] = 1 & FIRCCSR[FIRCERR] = 1
  - SIRC trimming error detected. SIRCCSR[SIRCERR_IE] = 1 & SIRCCSR[SIRCERR] = 1

21.4 External signals

Table 141. External signal descriptions

Signal	Description	Mode	I/O
EXTAL	SOSC Crystal input pin	SOSC	I
CLKOUT	SCG clock output pin	SCG	O

21.5 Initialization

Out of reset, the FIRC is default enabled. FIRCCFG[FREQ_SEL] is set to 010b, and RCCR[SCS] is set to 011b, so the FRO_HF clock which is set to 48 MHz would be as the system clock source.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Out of reset, the SIRC is default enabled.

Out of reset, all the other clock sources are disabled.

21.6 Application information

21.6.1 SOSC configuration example

1. Write 0 to SOSCCSR[LK] to unlock SOSCCSR
2. Write 1 to SOSCCSR[SOSCCM] to enable SOSC clock monitor
3. Read SOSCCSR[SOSCVLD] until it returns 1, indicating SOSC is valid
4. Read SOSCCSR[SOSCERR] to ensure it returns 0
5. Write 1 to RCCR[SCS] to switch main clock to SOSC
6. Read CSR[SCS] until it returns 1, indicating the switch is complete

21.6.2 SIRC configuration example

21.6.2.1 SIRC normal configuration example

1. Write 0 to SIRCCSR[LK] to unlock SIRCCSR
2. Write 1 to SIRCCSR[SIRCSTEN] to enable SIRC in Deep Sleep mode, if needed
3. Write 1 to SIRCCSR[SIRC_CLK_PERIPH_EN] to enable SIRC clock for peripheral use
4. Read SIRCCSR[SIRCVLD] until it returns 1, indicating SIRC is valid
5. Read SIRCCSR[SIRCERR] to ensure it returns 0
6. Write 2 to RCCR[SCS] to switch main clock to SIRC
7. Read CSR[SCS] until it returns 2, indicating the switch is complete
8. Write 1 to SIRCCSR[LK] to lock SIRCCSR

21.6.2.2 SIRC auto trim example

1. Follow steps 1 through 8 in ROSC configuration example
2. Write 3 to SIRCTCFG[TRIMSRC] to select ROSC as auto trim clock source
3. Write 0 to SIRCCSR[LK] to unlock SIRCCSR
4. Write 1 to SIRCCSR[SIRCTREN] to enable auto trim
5. Write 1 to SIRCCSR[SIRCTRUP] to enable update
6. Read SIRCCSR[SIRCVLD] until it returns 1, indicating SIRC is valid
7. Read SIRCCSR[SIRCERR] to ensure it returns 0
8. Read SIRCCSR[TRIM_LOCK] until it returns 1.
9. Write 1 to SIRCCSR[LK] to lock SIRCCSR

The auto trim process continues until it is disabled. The SIRCSTAT register shows the locked trim values. If there is an unlock after lock, SIRCCSR[SIRCERR] is set.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

21.6.3 FIRC configuration example

21.6.3.1 FIRC normal configuration example

1. Write 111 to FIRCCFG[FREQ_SEL] to select 144 MHz
2. Write 0 to FIRCCSR[LK] to unlock FIRCCSR
3. Write 0 to FIRCCSR[FIRC_FCLK_PERIPH_EN] to disable FIRC_HF clock for peripheral use, by default FIRC_HF is enabled
4. Write 1 to FIRCCSR[FIRC_SCLK_PERIPH_EN] to enable FIRC 36 MHz clock for peripheral use if needed
5. Write 1 to FIRCCSR[FIRCSTEN] to enable FIRC clock source in Deep Sleep mode if needed
6. Read FIRCCSR[FIRCVLD] until it returns 1, indicating FIRC is valid
7. Read FIRCCSR[FIRCERR] to ensure it returns 0
8. Write 3 to RCCR[SCS] to switch main clock to FIRC
9. Read CSR[SCS] until it returns 3, indicating the switch is complete
10. Write 0 to FIRCCSR[FIRCEN] to disable FIRC, by default FIRC is enabled
11. Write 1 to FIRCCSR[LK] to lock FIRCCSR

21.6.4 ROSC configuration example

1. Write 0 to ROSCCSR[LK] to unlock ROSCCSR
2. Write 1 to ROSCCSR[ROSCCM] to enable ROSC clock monitor if needed
3. Read ROSCCSR[ROSCVLD] until it returns 1, indicating ROSC is valid
4. Read ROSCCSR[ROSCERR] to ensure it returns 0
5. Write 4 to RCCR[SCS] to switch main clock to ROSC
6. Read CSR[SCS] until it returns 4, indicating the switch is complete
7. Write 1 to ROSCCSR[LK] to lock ROSCCSR

21.7 Memory map and register definition

This section includes information about the memory map and register definitions:

- Accesses on non-implemented registers that are outside the SCG maximum illegal address range generate a transfer error.
- Read accesses on non-implemented registers that are inside the SCG maximum illegal address range do not generate a transfer error.
- Write accesses on non-implemented registers that are inside the SCG maximum illegal address range generate a transfer error.
- Write accesses on read-only registers generate a transfer error.
- Only 32-bit writes are allowed for any writable SCG registers. Writes of 8 or 16 bits result in transfer errors.

21.7.1 SCG register descriptions

21.7.1.1 SCG memory map

SCG0 base address: 4008_F000h

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Offset	Register	Width (In bits)	Access	Reset value
0h	Version ID (VERID)	32	R	0000_0000h
4h	Parameter (PARAM)	32	R	0000_001Eh
8h	Trim Lock (TRIM_LOCK)	32	RW	0000_0000h
10h	Clock Status (CSR)	32	R	0300_0000h
14h	Run Clock Control (RCCR)	32	RW	0300_0000h
100h	SOSC Control Status (SOSCCSR)	32	RW	0000_0001h
108h	SOSC Configuration (SOSCCFG)	32	R	0000_0000h
200h	SIRC Control Status (SIRCCSR)	32	RW	0100_0020h
20Ch	SIRC Trim Configuration (SIRCTCFG)	32	RW	0000_0000h
210h	SIRC Trim (SIRCTRIM)	32	RW	See section
218h	SIRC Auto-trimming Status (SIRCSTAT)	32	RW	See section
300h	FIRC Control Status (FIRCCSR)	32	RW	See section
308h	FIRC Configuration (FIRCCFG)	32	RW	0000_0005h
310h	FIRC Trim (FIRCTRIM)	32	RW	See section
400h	ROSC Control Status (ROSCCSR)	32	RW	0100_0000h

21.7.1.2 Version ID (VERID)

Offset

Register	Offset
VERID	0h

Function

Indicates the version integrated for this instance on the device.

NOTE

Writing to this register results in a transfer error.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	VERSION															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	VERSION															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-0 VERSION	SCG Version Number This read-only field returns the SCG module version number.

21.7.1.3 Parameter (PARAM)

Offset

Register	Offset
PARAM	4h

Function

Indicates the feature parameters for this instance on the device.

NOTE

Writing to this register results in a transfer error.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0				0	0	0	0	0	0	0	ROSCCLK...	FIRCCLK...	SIRCCLK...	SOSCCLK...	0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Fields

Field	Function
31-11 —	Reserved
10 —	Reserved
9 —	Reserved
8 —	Reserved
7 —	Reserved
6 —	Reserved
5 —	Reserved
4 ROSCCLKPRES	ROSC Clock Present Indicates that the ROSC (FRO16K) clock source is present. 0b - ROSC clock source is not present 1b - ROSC clock source is present
3 FIRCCLKPRES	FIRC Clock Present Indicates that the FIRC clock source is present. 0b - FIRC clock source is not present 1b - FIRC clock source is present
2 SIRCCLKPRES	SIRC Clock Present Indicates that the SIRC clock source is present. 0b - SIRC clock source is not present 1b - SIRC clock source is present
1 SOSCCLKPRES	SOSC Clock Present Indicates that the SOSC clock source is present. 0b - SOSC clock source is not present 1b - SOSC clock source is present

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
0 —	Reserved

21.7.1.4 Trim Lock (TRIM_LOCK)

Offset

Register	Offset
TRIM_LOCK	8h

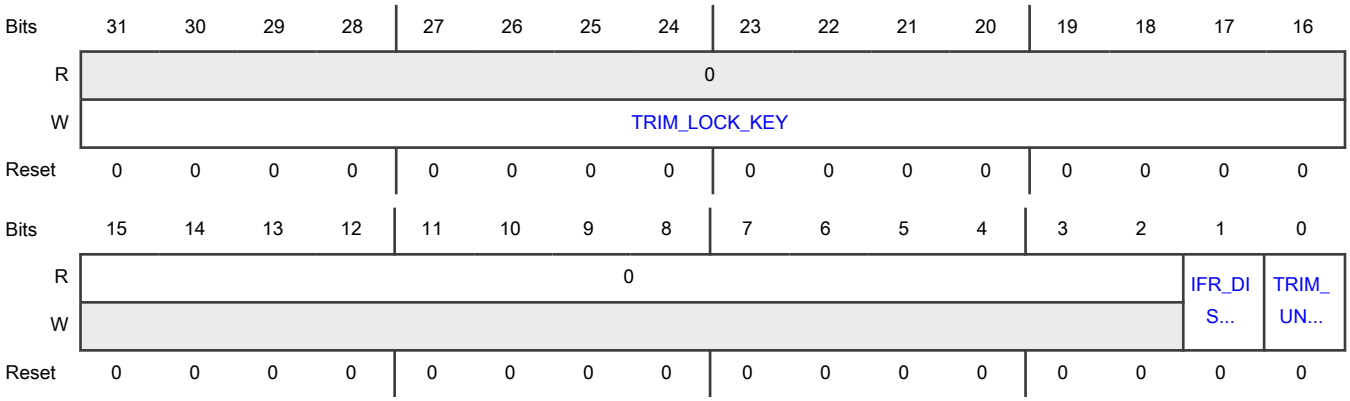
Function

Contains the TRIM_UNLOCK and IFR_DISABLE bits.

NOTE

The bits reset solely on a POR/LVD event.

Diagram



Fields

Field	Function
31-16 TRIM_LOCK_KEY	TRIM_LOCK_KEY Write value that is used to unlock writes to the IFR_DISABLE and TRIM_UNLOCK registers. **NOTE** A write value of 0x5a5a allows writes to the IFR_DISABLE and TRIM_UNLOCK to take affect.

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	NOTE Writable but reads always return a 0 value.
15-2 —	Reserved
1 IFR_DISABLE	IFR_DISABLE Locks IFR write access to SCG trim registers. When set, prevents SCG trim registers from getting IFR loaded during warm reset. **NOTE** IFR_DISABLE is writable during write access to TRIM_LOCK and only when the TRIM_LOCK[31:16] write value is equal to 16'h5a5a. 0b - IFR write access to SCG trim registers not disabled. The SCG Trim registers are reprogrammed with the IFR values after any system reset. 1b - IFR write access to SCG trim registers during system reset is blocked.
0 TRIM_UNLOCK	TRIM_UNLOCK Locks user writes access to SCG Trim Configuration registers. **NOTE** TRIM_UNLOCK is writable during write access to TRIM_LOCK and only when the TRIM_LOCK[31:16] write value is equal to 16'h5a5a. 0b - SCG Trim Registers locked and not writable. 1b - SCG Trim registers unlocked and writable.

21.7.1.5 Clock Status (CSR)

Offset

Register	Offset
CSR	10h

Function

Shows the currently configured clock source that is generating the system clock.

NOTE

Writing to the Clock Status Register (CSR) results in a transfer error.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0					SCS			0							
W																
Reset	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-27 —	Reserved
26-24 SCS	System Clock Source Returns the currently configured clock source generating the system clock. Reports the configuration set by the RCCR. 001b - SOSC (CLK_IN) 010b - SIRC (FRO12M) 011b - FIRC (FRO_HF) 100b - ROSC (FRO_16K) All other values are reserved.
23-0 —	Reserved

21.7.1.6 Run Clock Control (RCCR)

Offset

Register	Offset
RCCR	14h

Function

Selects the clock source generating the system clock.

NOTE

Switching to new system clock source must be done after previous RCCR changes have been completed and the Clock Status Register has been updated to match the present RCCR setting.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0					SCS			0							
W																
Reset	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-27 —	Reserved
26-24 SCS	System Clock Source Selects the clock source generating the system clock in Run mode. Attempting to select a clock that is not valid is ignored. Selecting a different clock source when in Run mode requires that clock source to be enabled first and be valid before system clocks are allowed to switch to that clock source. **NOTE** Programming SCS to a different value must only be done after the previous SCS clock switch has completed. 001b - SOSC (CLK_IN) 010b - SIRC (FRO12M) 011b - FIRC (FRO_HF) 100b - ROSC (FRO_16K) All other values are reserved.
23-0 —	Reserved

21.7.1.7 SOSC Control Status (SOSCCSR)

Offset

Register	Offset
SOSCCSR	100h

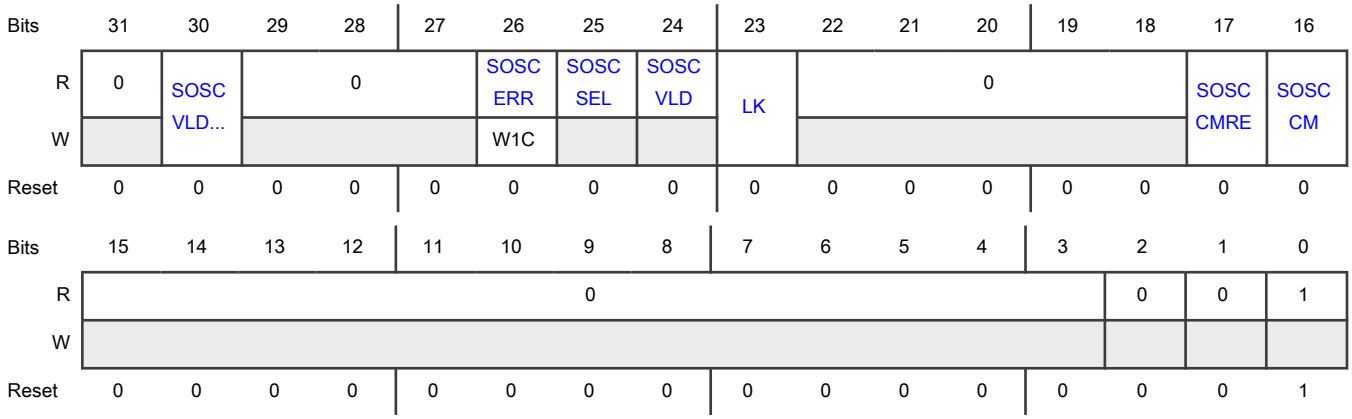
Function

Contains control and status bits for the SOSC clock source.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram



Fields

Field	Function
31 —	Reserved
30 SOSCVLD_IE	SOSC Valid Interrupt Enable Generates an interrupt when SOSCVLD is asserted. 0b - SOSCVLD interrupt is not enabled 1b - SOSCVLD interrupt is enabled
29-27 —	Reserved
26 SOSCERR	SOSC Clock Error This flag is reset on Chip POR only, software can also clear this flag by writing a logic one. 0b - SOSC Clock Monitor is disabled or has not detected an error 1b - SOSC Clock Monitor is enabled and detected an error
25 SOSCSEL	SOSC Selected This flag shows the SOSC clock source is selected as the system clock source. 0b - SOSC is not the system clock source 1b - SOSC is the system clock source
24 SOSCVLD	SOSC Valid This flag shows the SOSC clock source is valid. 0b - SOSC is not enabled or clock is not valid 1b - SOSC is enabled and output clock is valid
23	Lock

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
LK	Locks this register so that it cannot be written to. **NOTE** This field can be cleared/set at any time. 0b - This Control Status Register can be written 1b - This Control Status Register cannot be written
22-18 —	Reserved
17 SOSCCMRE	SOSC Clock Monitor Reset Enable Enables the SOSCERR generate reset. 0b - Clock monitor generates an interrupt when an error is detected 1b - Clock monitor generates a reset when an error is detected
16 SOSCCM	SOSC Clock Monitor Enable Enables the clock monitor when SOSCVLD is set. **NOTE** SOSC clock monitor remains enabled in Deep Sleep mode only if SOSCCM is enabled. If the clock is disabled in Deep Sleep mode, then SOSCCM must be cleared to prevent unexpected SOSC loss of clock. The clock monitor is always disabled in Power Down and Deep Power Down modes. When the clock monitor is disabled in a low power mode, it remains disabled until the clock valid flag is set, following exit from the low power mode. **NOTE** The reference clock used to monitor the SOSC is the SIRC. This clock must be programmed to be enabled in order to monitor the SOSC. SIRC is automatically disabled in Power Down and Deep Power Down modes and the clock monitor of SOSC is disabled. 0b - SOSC Clock Monitor is disabled 1b - SOSC Clock Monitor is enabled
15-3 —	Reserved
2 —	Reserved
1 —	Reserved
0 —	Reserved

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

21.7.1.8 SOSC Configuration (SOSCCFG)

Offset

Register	Offset
SOSCCFG	108h

Function

Contains the clock frequency range select and external reference select for the SOSC clock source.

NOTE

The SOSCCFG register cannot be changed when the SOSC is enabled. When the SOSC is enabled, writes to this register are ignored, and there is no transfer error.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0				0				0		0		0	EREFS	0	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-12 —	Reserved
11-8 —	Reserved
7-6 —	Reserved
5-4 —	Reserved
3 —	Reserved

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
2 EREFS	External Reference Select Selects the source for the external reference clock. This bit selects which clock is output from the SOSC into the SCG, whether from the crystal oscillator or from an external clock input. 0b - External reference clock selected. 1b - Internal crystal oscillator of OSC selected.
1-0 —	Reserved

21.7.1.9 SIRC Control Status (SIRCCSR)

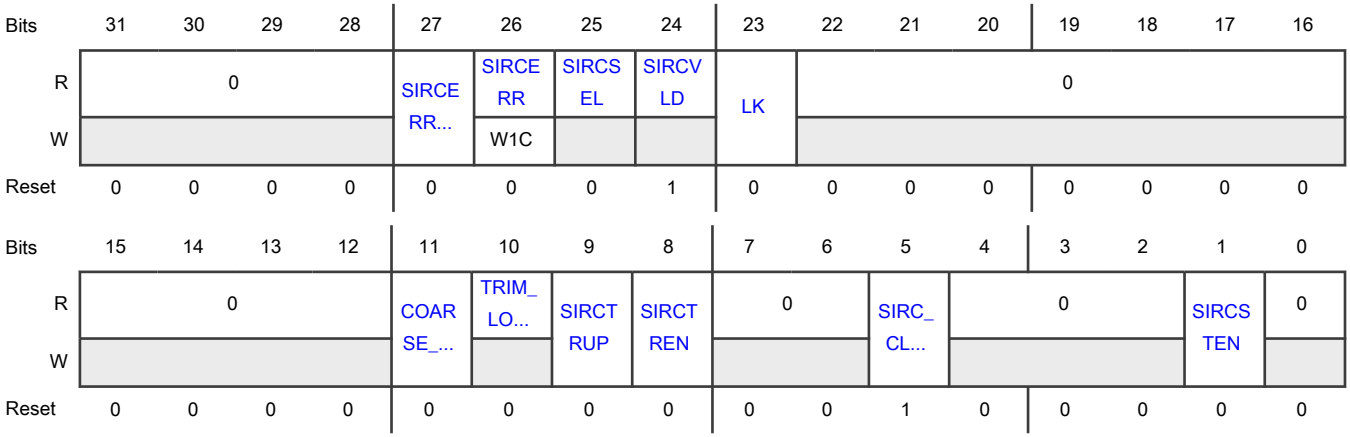
Offset

Register	Offset
SIRCCSR	200h

Function

Contains control and status bits for the SIRC clock source.

Diagram



Fields

Field	Function
31-28 —	Reserved
27	SIRC Clock Error Interrupt Enable

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
SIRCERR_IE	Generates an interrupt when SIRCERR is asserted. 0b - SIRCERR interrupt is not enabled 1b - SIRCERR interrupt is enabled
26 SIRCERR	SIRC Clock Error This flag is reset on Chip POR only, software can also clear this flag by writing a logic one 0b - Error not detected with the SIRC trimming 1b - Error detected with the SIRC trimming
25 SIRCSEL	SIRC Selected This flag shows the SIRC clock source is selected as the system clock source. 0b - SIRC is not the system clock source 1b - SIRC is the system clock source
24 SIRCVLD	SIRC Valid This flag shows the SIRC clock source is valid. 0b - SIRC is not enabled or clock is not valid 1b - SIRC is enabled and output clock is valid
23 LK	Lock This field can be cleared/set at any time. The purpose for this bitfield is to prevent runaway code from changing SIRC clock configurations. 0b - Control Status Register can be written 1b - Control Status Register cannot be written
22-12 —	Reserved
11 COARSE_TRIM_BYPASS	Coarse Auto Trim Bypass This bit is to bypass the coarse trim step when auto-trimming. 0b - SIRC Coarse Auto Trim NOT Bypassed 1b - SIRC Coarse Auto Trim Bypassed
10 TRIM_LOCK	SIRC TRIM LOCK When SIRCTREN and SIRCTRUP are enabled, TRIM_LOCK indicates when auto trimming is complete and output SIRC frequency has locked to target SIRC range.

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	NOTE • This field is automatically cleared if SIRCTREN and SIRCTRUP are not set. • TRIM_LOCK is not set when SIRCTCFG[TRIMSRC] = 2'b00, i.e., using full speed USB as the trim source. 0b - SIRC auto trim not locked to target frequency range 1b - SIRC auto trim locked to target frequency range
9 SIRCTRUP	SIRC Trim Update This bit allows the SIRCSTAT to be updated by the auto-trimming hardware. 0b - Disables SIRC trimming updates 1b - Enables SIRC trimming updates
8 SIRCTREN	SIRC 12 MHz Trim Enable (SIRCCFG[RANGE]=1) This bit enables the auto trim of SIRC 12 MHz by an external clock source. 0b - Disables trimming SIRC to an external clock source 1b - Enables trimming SIRC to an external clock source
7-6 —	Reserved
5 SIRC_CLK_PERIPH_EN	SIRC Clock to Peripherals Enable This bit enables the SIRC clock for peripheral use. 0b - SIRC clock to peripherals is disabled 1b - SIRC clock to peripherals is enabled
4-2 —	Reserved
1 SIRCSTEN	SIRC Stop Enable Enables the SIRC clock source in Deep Sleep mode. 0b - SIRC is disabled in Deep Sleep mode 1b - SIRC is enabled in Deep Sleep mode
0 —	Reserved

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

21.7.1.10 SIRC Trim Configuration (SIRCTCFG)

Offset

Register	Offset
SIRCTCFG	20Ch

Function

Contains the auto trim clock source select and trim clock divider control for the SIRC clock source.

NOTE

The SIRCTCFG register cannot be changed when SIRC tuning is enabled. When the SIRC tuning is enabled, writes to this register are ignored, and there is no transfer error.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0								TRIMDIV							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0				0				0						TRIMSRC	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-23 —	Reserved
22-16 TRIMDIV	SIRC Trim Pre-divider Divider of SOSC for SIRC trimming. **NOTE** When selecting SOSC as SIRC trimming source the TRIMDIV register must be set to correct div ratio to generate 1 MHz output reference trimming clock. **NOTE** TRIMDIV is an N-Divider supporting a div ratio of 1 (TRIMDIV=0x00) to a div ratio of 128 (TRIMDIV=0x7f).
15-11	Reserved

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
—	
10-8 —	Reserved
7-2 —	Reserved
1-0 TRIMSRC	Trim Source Configures the external clock source to tune the SIRC. TRIMSRC must be configured before programming SIRCSTAT register for trim update 00b - Reserved 01b - Reserved 10b - SOSC. This option requires that SOSC be divided using the TRIMDIV field to get a frequency of 1 MHz. 11b - ROSC (16.384 KHz)

21.7.1.11 SIRC Trim (SIRCTRIM)

Offset

Register	Offset
SIRCTRIM	210h

Function

Holds the trim values for the SIRC clock source. This register is loaded from IFR during reset.

NOTE

This register writes are protected by TRIM LOCK register.

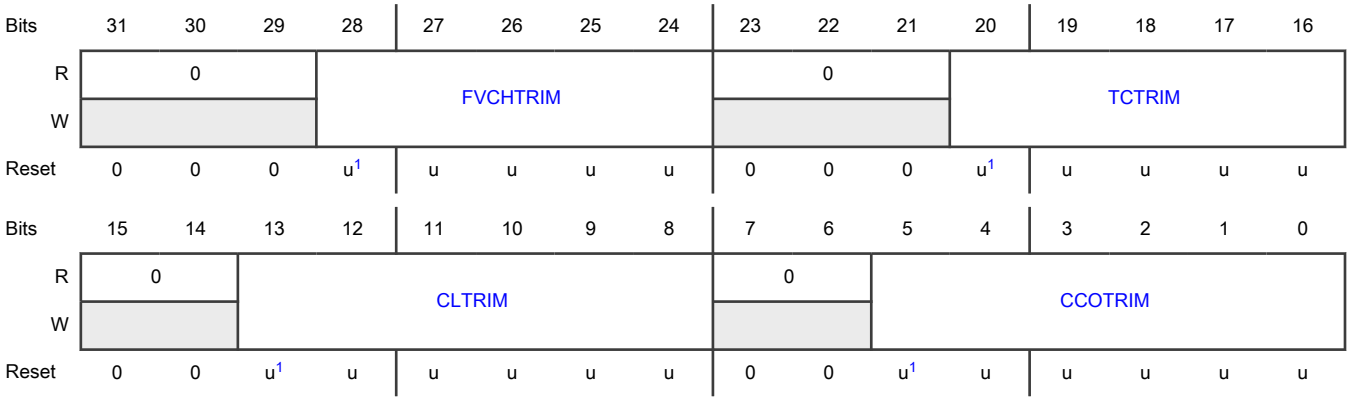
NOTE

It is not recommended to manually modify the values in this register because the relationship between the trim value and frequency is not linear, so the values are not intended for manual adjustment.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram



1. Reset values are loaded out of IFR.

Fields

Field	Function
31-29 —	Reserved
28-24 FVCHTRIM	Calibrates the replica voltage in FSU for CCO to get well frequency at initial period
23-21 —	Reserved
20-16 TCTRIM	Trim Temp Trim bus to calibrate the relationship between freq and temperature.
15-14 —	Reserved
13-8 CLTRIM	CL Trim Trims bus to calibrate the freq of CCO in close loop mode to within approximately ±0.6% of the target 12 MHz frequency.
7-6 —	Reserved
5-0 CCOTRIM	CCO Trim Trims bus to calibrate the freq of CCO in open loop mode to within approximately ±2.5% of the target 12 MHz frequency.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

21.7.1.12 SIRC Auto-trimming Status (SIRCSTAT)

Offset

Register	Offset
SIRCSTAT	218h

Function

This register is loaded from IFR during warm reset.

When SIRCTREN=1 and SIRCTRUP=1, this register is uploaded with the trim values generated by SIRC auto-trimming which is enabled.

When SIRCTREN=1 and SIRCTRUP=0, manually writing to this register is not recommended, as these values are obtained through the autotrim function and used to trim SIRC clock.

NOTE

TRIMSRC needs to be programmed to TRIMSRC=10 , writes to this register are allowed and values written to this register are used to trim the SIRC clock.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		CLTRIM						0		CCOTRIM					
W																
Reset	0	0	u[1]	u	u	u	u	u	0	0	u[1]	u	u	u	u	u

1. Reset values are loaded out of IFR.

Fields

Field	Function
31-14 —	Reserved
13-8 CLTRIM	CL Trim CLTRIM bits are to calibrate the freq of CCO in close loop mode, are used to trim the SIRC Clock to within approximately ±0.6 % of the target 12 MHz frequency.
7-6 —	Reserved

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
5-0 CCOTRIM	CCO Trim CCOTRIM bits are to calibrate the freq of CCO in open loop mode, are used to coarsely trim the SIRC Clock to within approximately ±2.5 % of the target 12 MHz frequency.

21.7.1.13 FIRC Control Status (FIRCCSR)

Offset

Register	Offset
FIRCCSR	300h

Function

Contains control and status bits for the FIRC clock source.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	FIRCACC	FIRCACC...	0		FIRCERR...	FIRCERR	FIRCSEL	FIRCVLD	LK	0			0			
W						W1C										
Reset	u	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0				0				0		FIRC_FC...	FIRC_SC...	0		FIRCSTEN	FIRCEN
W																
Reset	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	1

Fields

Field	Function
31 FIRCACC	FIRC Frequency Accurate This flag shows the FIRC clock source is accurate. 0b - FIRC is not enabled or clock is not accurate. 1b - FIRC is enabled and output clock is accurate after some preparation time which is obtained by counting FRO_HF clock.
30 FIRCACC_IE	FIRC Accurate Interrupt Enable Generate an interrupt when FIRCACC is asserted.

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	0b - FIRCACC interrupt is not enabled 1b - FIRCACC interrupt is enabled
29-28 —	Reserved
27 FIRCERR_IE	FIRC Clock Error Interrupt Enable Generate an interrupt when FIRCERR is asserted. 0b - FIRCERR interrupt is not enabled 1b - FIRCERR interrupt is enabled
26 FIRCERR	FIRC Clock Error This flag is reset on Chip POR only, software can also clear this flag by writing a logic one 0b - Error not detected with the FIRC trimming 1b - Error detected with the FIRC trimming
25 FIRCSEL	FIRC Selected This flag shows the FIRC clock source is selected as the system clock source. 0b - FIRC is not the system clock source 1b - FIRC is the system clock source
24 FIRCVLD	FIRC Valid status This flag shows the FIRC clock source is valid. 0b - FIRC is not enabled or clock is not valid. 1b - FIRC is enabled and output clock is valid. The clock is valid after there is an output clock from the FIRC analog.
23 LK	Lock This bit field can be cleared/set at any time. The purpose for this bitfield is to prevent runaway code from changing FIRC clock configurations. 0b - Control Status Register can be written 1b - Control Status Register cannot be written
22-20 —	Reserved
19-16 —	Reserved
15-12	Reserved

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
—	
11-8 —	Reserved
7-6 —	Reserved
5 FIRC_FCLK_PERIPH_EN	FRO_HF_GATED to Peripherals Enable Enables the FRO_HF_GATED signal for peripheral use. 0b - FRO_HF_GATED disabled 1b - FRO_HF_GATED enabled
4 FIRC_SCLK_PERIPH_EN	FIRC_36MHz Clock to Peripherals Enable Enables the FIRC_36MHz clock for peripheral use. 0b - FIRC_36MHz disabled 1b - FIRC_36MHz enabled
3-2 —	Reserved
1 FIRCSTEN	FIRC Stop Enable Enables the FIRC clock source in Deep Sleep mode if FIRCEN is set. 0b - FIRC is disabled in Deep Sleep mode 1b - FIRC is enabled in Deep Sleep mode
0 FIRCEN	FIRC Enable Enables the FIRC clock source. 0b - FIRC is disabled 1b - FIRC is enabled

21.7.1.14 FIRC Configuration (FIRCCFG)

Offset

Register	Offset
FIRCCFG	308h

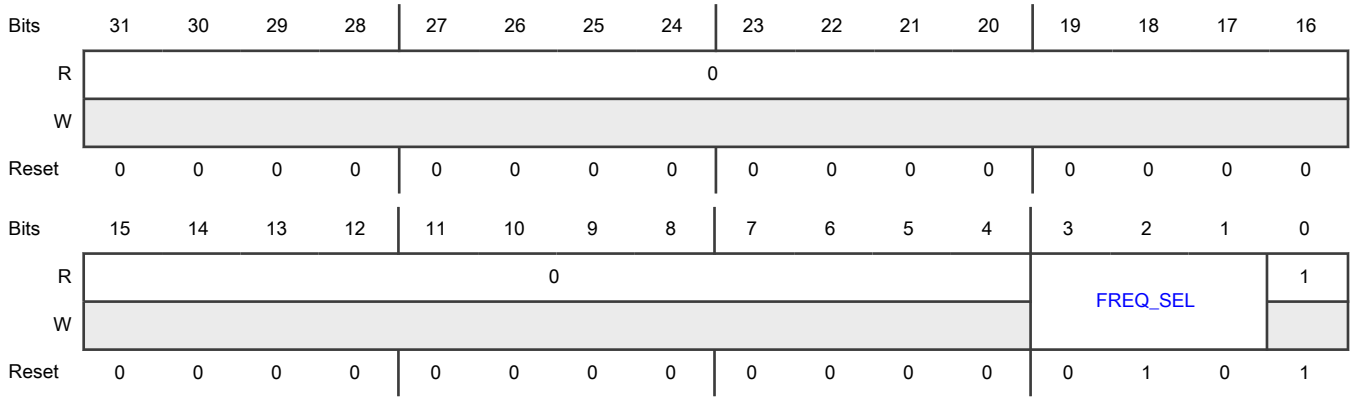
Function

Controls the clock frequency range select for the FIRC clock source.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram



Fields

Field	Function
31-4 —	Reserved
3-1 FREQ_SEL	Frequency select These bits are used to select the frequency for FRO_HF clock when used as system clock. 000b - 36 MHz FIRC clock selected 010b - 48 MHz FIRC clock selected, divided from 144 MHz 100b - 72 MHz FIRC clock selected 110b - 144 MHz FIRC clock selected
0 —	Reserved

21.7.1.15 FIRC Trim (FIRCTRIM)

Offset

Register	Offset
FIRCTRIM	310h

Function

Holds the trim values for the FIRC clock source. This register is loaded from IFR during reset. These values are used for trimming the different selected frequencies (from 36 to 144 MHz by FIRCCFG[FREQ_SEL]) of FIRC.

NOTE

This register writes are protected by TRIM LOCK register.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

NOTE

It is not recommended to manually modify the values in this register because the relationship between the trim value and frequency is not linear, so the values are not intended for manual adjustment.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0		TRIMSTART						0				TRIMTEMP			
W																
Reset	0	0	u[1]	u	u	u	u	u	0	0	0	0	u[1]	u	u	u

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		TRIMCOAR						TRIMFINE							
W																
Reset	0	0	u[1]	u	u	u	u	u	u[1]	u	u	u	u	u	u	u

1. Reset values are loaded out of IFR.

Fields

Field	Function
31-30 —	Reserved
29-24 TRIMSTART	Trim Start Current dac adjustment of the replica cco current of start-up circuit.
23-20 —	Reserved
19-16 TRIMTEMP	Trim Temperature Temperature coefficient compensation for FIRC .
15-14 —	Reserved
13-8 TRIMCOAR	Trim Coarse TRIMCOAR bits are used to coarsely trim the FIRC Clock to within approximately ±3.2% of the target frequency.
7-0 TRIMFINE	Trim Fine Current DAC adjustment of the CCO current. TRIMFINE bits are used to fine trim the FIRC Clock to within approximately ±0.25% of the target frequency.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

21.7.1.16 ROSC Control Status (ROSCCSR)

Offset

Register	Offset
ROSCCSR	400h

Function

Contains control and status bits for the ROSC (FRO16K) clock source.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0					ROSCERR	ROSCSEL	ROSCVLD	LK	0					ROSCCMRE	ROSCCM
W						W1C										
Reset	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-27 —	Reserved
26 ROSCERR	ROSC Clock Error This flag is reset on Chip POR only, software can also clear this flag by writing a logic one 0b - ROSC Clock Monitor is disabled or has not detected an error 1b - ROSC Clock Monitor is enabled and has detected an error
25 ROSCSEL	ROSC Selected This flag shows the ROSC clock source is selected as the system clock source. 0b - ROSC is not the system clock source 1b - ROSC is the system clock source
24 ROSCVLD	ROSC Valid This flag shows the ROSC clock source is valid. 0b - ROSC is not enabled or clock is not valid

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	1b - ROSC is enabled and output clock is valid
23 LK	Lock Locks this register so that it cannot be written to. **NOTE** This field can be cleared/set at any time. 0b - Control Status Register can be written 1b - Control Status Register cannot be written
22-18 —	Reserved
17 ROSCCMRE	ROSC Clock Monitor Reset Enable Enables the ROSCERR to generate a reset. 0b - Clock monitor generates an interrupt when an error is detected 1b - Clock monitor generates a reset when an error is detected
16 ROSCCM	ROSC Clock Monitor Enables the clock monitor when ROSCVLD is set. **NOTE** When the clock monitor is disabled in a low power mode, it remains disabled until the clock valid flag is set following exit from the low power mode. **NOTE** The reference clock used to monitor the ROSC is the SIRC. The SIRC clock must be programmed to be enabled in order to monitor the ROSC. SIRC is automatically disabled in Power Down and Deep Power Down modes, and the clock monitor of the ROSC is disabled. 0b - ROSC clock monitor is disabled 1b - ROSC clock monitor is enabled
15-0 —	Reserved

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Chapter 22
Power Management

22.1 Introduction

This chapter describes the power modes supported with this device. It also describes the power domains and the voltage supply options connected to these power domains. For details about changing power modes and configuring those modes, see the CMC chapter.

22.2 Power domains

A power domain is a collection of circuits that can be powered off even when a voltage source is still supplied. This section shows the power domains within the device and the placement of the modules within these domains.

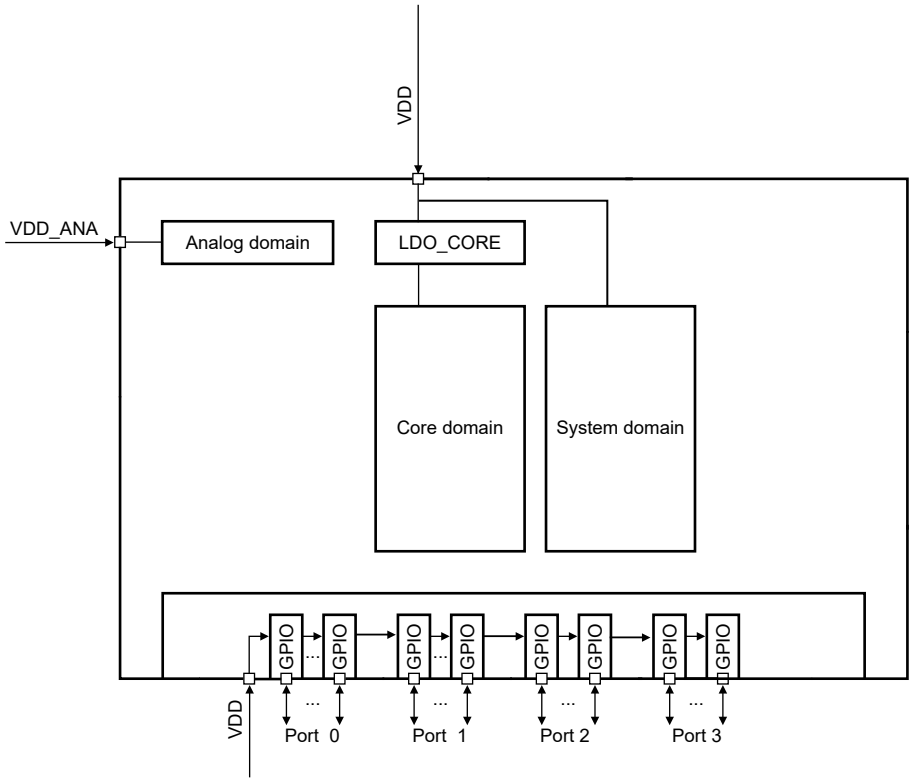


Figure 76. Power domains and voltage supplies

Table 142. Power Domain

Domain	Voltage supply	Module
CORE	LDO_CORE	CM23
		NVIC
		Systick Timer
		Mutilayer AHB Matrix
		PBridge0/1

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table 142. Power Domain (continued)

		DAP
		Debug Mailbox
		SWJ
		CMC
		SECCON
		CRC0
		DMA0
		SYSCON
		MBC
		FMC0
		FMU0
		Flash
		FRO12M (Analog Circuitry)
		FRO144M (Analog Circuitry)
		SCG
		LPI2C0
		LPSPI0
		LPUART0-3
		ADC0 (Digital Circuitry)
		CTIMER0-1
		Frequency Measurement Unit
		PWM0
		WWDT0/1
		ADC0-GP (Digital Circuitry)
		CMP0 (Digital Circuitry)
		OPAMP0 (Digital Circuitry)
		GPIO0-3
		PORT0-3
SRAM	LDO_CORE / RAM_RET_LDO	SRAM
SYSTEM	VDD	SPC0
		HVD/LVD/POR (Analog Circuitry)
		LDO_CORE (Analog Circuitry)
		RAM_RET_LDO (Analog Circuitry)

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table 142. Power Domain (continued)

		VBAT
		FRO16K (Analog Circuitry)
		WUU
		LPTMR0
		OSC32K (analog circuitry)
		CMP0 (Analog Circuitry)
		PORT0/1/2/3 (Pins)
ANALOG	VDD_ANA	ADC0 (Analog Circuitry)
		OPAMP0 (Analog Circuitry)

22.3 Power modes

The device supports Active, Sleep, Deep Sleep, Power Down and Deep Power Down. Any reset brings the chip back to the Active mode.

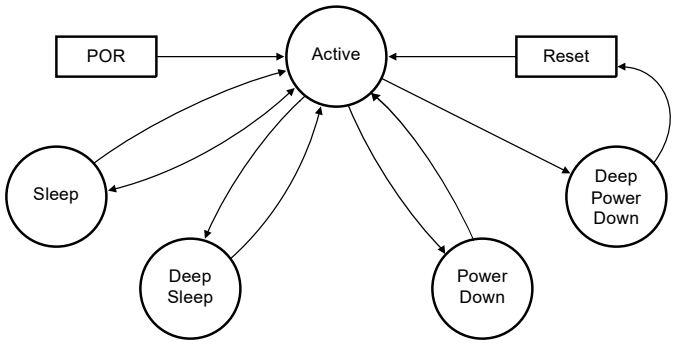


Figure 77. Power modes

22.3.1 Active mode

In Active mode, CPU execution is possible.

To achieve the performance requirements of a given system application while minimizing power consumption, Active mode permits the following power-saving options when possible:

- Configure the VDD_CORE subdomains voltage level to balance power and performance.
- Gate the clocks to unused modules.

22.3.2 Sleep mode

In the Sleep mode, the CPU0 clock is off, and the system clock and bus clock remain ON. Most modules can remain operational. To achieve the performance requirements of a given system application while minimizing power consumption, the Sleep mode permits the following power-saving options when possible:

- Configure the VDD_CORE sub-domains voltage level to balance power and performance.
- Gate the clocks to unused modules.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

22.3.3 Deep Sleep mode

In Deep Sleep mode, CPU execution is halted. The core clock is gated off.

The Deep Sleep mode supports the following behaviors based on different clock configurations:

- CPU clock, system clock and bus clock are all OFF.
- Some modules can remain operational with low power asynchronous clock sources and serve as wake-up sources.
- System RAM is configured to state retention mode.
- SRAM is in deep sleep mode.

Wake from Deep Sleep mode is triggered by an interrupt or a wakeup event.

The Deep Sleep mode also supports a partial wake-up where a bus master other than the CPU (eg. DMA) is recovered by a wakeup event. The device automatically re-enters the Deep Sleep mode after this module finishes its task.

To achieve the performance requirements of a given system application while minimizing power consumption, the Deep Sleep mode permits the following power-saving options when possible:

- Configure the VDD_CORE subdomains voltage level to balance power and performance.
- Gate the clocks to unused modules.

22.3.4 Power Down mode

Power down mode places CORE domain of the chip into a static state. It is the lowest power mode that can retain all registers. External signals can wake the chip via SYSTEM domain peripherals.

To enter power down mode, CMC PMCTRLMAIN[LPMODE] should be configured to 0011b, and SPC LP_CFG[CORELDO_VDD_LVL] should be configure 00b.

Flash memory is powered off. SRAM is in deep sleep mode.

The power down mode also supports a partial wake-up where a bus initiator other than the CPU is recovered by a wakeup event in SYSTEM domain. The device automatically re-enters the Power down mode after this module finishes its task.

22.3.5 Deep Power Down mode

The device wakes from Deep Power Down mode through the Reset routine.

In Deep Power Down mode, the whole CORE domain (including all subdomains) is power gated.

In Deep Power Down mode, the on-chip regulator for VDD_CORE is powered off. The peripherals in SYSTEM domain can be still alive.

External reset or the SYSTEM domain peripherals can wake the device.

SRAMA can be retained optionally in Deep Power Down mode.

22.4 Module operation in low power modes

The following table shows the functionality of each module in low power modes.

Module	Sleep	Deep Sleep	Power Down	Deep Power Down
Core Modules				
Multi-Layer AHB Matrix	On[1]	Static[2]	Static	Off[3]
CM23	Static	Static	Static	Off
System Modules				

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Module	Sleep	Deep Sleep	Power Down	Deep Power Down
CRC	On	Static	Static	Off
CMC	On	On	Static	Off
eDMA3	On	LP[4]/Static	Static	Off
EIM	On	Static	Static	Off
ERM	On	Static	Static	Off
MBC	On	Static	Static	Off
Peripheral Input Mux	On	On/Static	Static	Off
RMC	On	On	On	On
SCG-Lite	On	On/Static	Static	Off
SPC	On	On	On	On
SYSCON	On	On/Static	Static	Off
WUU	On	On	On	On
VBAT Wrapper	On	On	On	On
FRO144M	On/Off	On/Off	Off	Off
FRO12M	On	On/Off	Off	Off
FRO16K	On/Off	On/Off	On/Off	On/Off
OSC32K	On/Off	On/Off	On/Off	On/Off
PMC Sub Module				
BG	On	On/Off	Off	Off
LDO_CORE	On	LP/Off	LP	Off
PoR	On	On	On	On
RAM_Retention_LDO	On/Off	On/Off	On/Off	On/Off
VDD_LVD/HVD	On/Off	On	Off	Off
VDD_CORE_LVD	On/Off	On	Off	Off
Flash Array	On	Static	Off	Off
FMC	On	Static	Static	Off
FMU	On	Static	Static	Off
SRAM	On	Retention	Retention	Retention/Off
Timers				
CTIMER0-1	On	Static/LP	Static	Off
FREQME	On	Static/LP	Static	Off
FlexPWM0	On	Static	Static	Off

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Module	Sleep	Deep Sleep	Power Down	Deep Power Down
LPTMR0	On	Static/LP	Static/LP	Static/LP
WWDT0-1	On	Static/LP	Static	Off
RTC	On	Static/LP	Static/LP	Static/LP
Communication				
TRNG	On	Static	Static	Off
LPI2C0	On	Static/LP	Static	Off
LPSPI0	On	Static/LP	Static	Off
LPUART0-3	On	Static/LP	Static	Off
Analog				
16-bit ADC Digital	On	Static/LP	Static	Off
16-bit ADC Analog	On/Off	On/Off	Off	Off
CMP0 Digital	On	Static/LP	Static	Off
CMP0 Analog	On/Off	On/Off	On/Off	On/Off
OPAMP Digital	On	Static/LP	Static	Off
OPAMP Analog	On/Off	On/Off	Off	Off
HMI				
GPIO0-3	On	Static/LP	Static	Off
PORT0-3	On	Static	Static	Off
PORT0-3 Pins	On	On	Static/LP	Static/LP
Debug and Test				
SWD	On	Static	Static	Off

1. On means module is functional and accessible via memory map. For analog module, means it is enabled.
2. Static means the module is not active in state retention status.
3. Off means the module can be powered off.
4. LP means that the module is in low power state (clock gated, asynchronous operation, etc). For digital module, it can be active with async functional clock.

NOTE

User cannot disable the regulator which powers the chip in Power Down mode and higher power mode.

22.5 Power optimization

22.5.1 VDD_CORE voltage scaling

This device supports VDD_CORE voltage scaling for different performance and power consumption.

This device supports the following VDD_CORE voltage scaling methods:

In Active, Sleep, and Deep Sleep modes, VDD_CORE can be configured to 1.1V in SPC. When VDD_CORE is 1.1V, it is Standard Drive (SD) mode. Lower VDD_CORE in PD mode, can further reduce VDD_CORE level below 1.1V in SPC. The leakage can be

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

reduced. VDD_CORE domain can be retained, but all logics in VDD_CORE are static. CMC Power Mode Control Register must be configured to 4'b0011.

22.5.2 SRAM configuration

The SRAM is partitioned, allowing the mode of each SRAM partition to be independently configured:

SRAM power mode	Description
Active	SRAM can be accessed normally.
Deep Sleep	SRAM cannot be accessed, but the data is retained.
Shutdown	SRAM cannot be accessed, and the data is not retained.

Depending on the requirements of the system application, software can optionally configure an individual SRAM partition to be in a lower, power-saving mode relative to the overall device power mode:

When the overall device power mode is...	An SRAM partition may be configured into this mode...
Active/Sleep	Active
Deep Sleep	Deep Sleep

NOTE

SPC implements RAM_RET_LDO which can retain SRAM in Deep Power Down (DPD) mode. There are registers in SPC to control which SRAM can be retained in DPD mode.

22.5.3 Flash low power options

The internal flash memory can enter low power mode:

- in Active mode under software control (FLASHCR[FLASHDIS])
- optionally in Sleep mode (FLASHCR[FLASHDOZE])
- in Deep Sleep mode

NOTE

To avoid unexpected access latencies, take care when configuring the flash module to a low power mode when the overall device is in Active mode. The flash module needs time to recover full-speed functionality after entering a low power mode.

22.5.4 Peripheral clock gating

To conserve power, the clocks to modules can be turned off or divided by configuring the SYSCON module registers.

22.5.5 Voltage monitor optimization

Depending on the requirements of the system application, software can disable individual voltage monitors (LVD and/or HVD) within the different power domains to save power.

22.5.6 Wake-up time consideration

This device supports different low power modes for power consumption level and wake-up time balance.

The wake-up time from Sleep, Deep Sleep and Power Down mode is the accumulation of following time:

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

- VDD_CORE power source recovery time
- The longer of the following:
  - Clock recovery time
  - Flash recovery time
- Interrupt Latency

The wake-up time from Deep Power Down mode is:

- The boot-up time required after device released by POR. This includes everything included in a cold power-up, including the Boot ROM latency.

The clock recovery time can be affected by following factors:

- The system clock source status in low power modes. There will be no recovery time needed if the clock remains ON in low power mode.
- The system clock source before entering low power mode.

The flash recovery time depends on the flash status in low power mode.

The VDD_CORE power source recovery time can be affected by following factors:

- The regulator status in low power modes. There will be no recovery time needed if the regulator remains in full-power regulation in low power mode.
- Whether VDD_CORE voltage scaling is enabled before and after entering low power modes

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Chapter 23
VBAT

23.1 Chip-specific VBAT information

Table 143. Reference links to related information

Topic	Related module	Reference
Full description	VBAT	VBAT
System memory map		Memory map
Clocking		Clock distribution
Power Management		Power Management
Signal multiplexing	Port control	Signal multiplexing

23.1.1 Module instances

This chip contains one instance of the VBAT module, VBAT0.

23.1.2 Unsupported features

This chip only supports FRO16K feature of VBAT. All other features are not supported and references to RAM LDO, bandgap timer, and switch, should be ignored.

23.1.3 Connection for OSCCLKE[CLKE] and FROCLKE[CLKE]

Table 144. Connection for OSCCLKE[CLKE] and FROCLKE[CLKE]

VBAT clock enable	Connect to this power
CLKE[1]	CORE_MAIN
CLKE[0]	VDD

23.2 Overview

VBAT implements the 16 kHz internal clock source via FRO16K.

23.2.1 Block diagram

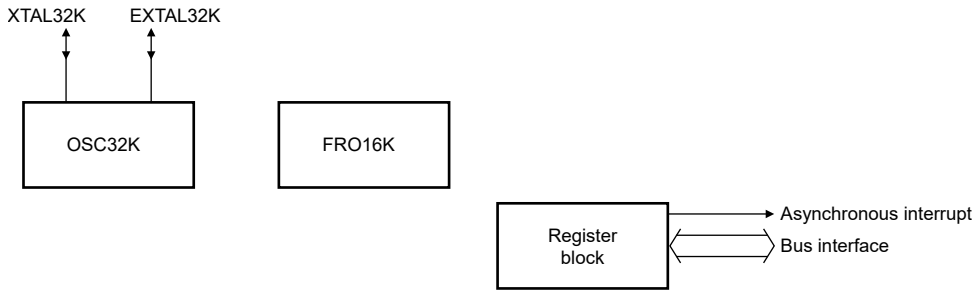


Figure 78. Block diagram

All information provided in this document is subject to legal disclaimers.
© 2026 NXP B.V. All rights reserved.

23.2.2 Features

- Includes oscillator for 32.768 kHz crystal (OSC32K)
- Supports internal 16 kHz free-running oscillator (FRO16K)

23.3 Functional description

The following sections describe the functional details of VBAT.

23.3.1 Operations

This section describes the operations of VBAT.

23.3.1.1 FRO 16.384 kHz clock

FRO16K is enabled by default on POR. You can disable it or lock it to prevent any change to the enable field.

23.3.1.2 OSC 32.768 kHz clock

OSC32K is disabled by default on VBAT POR. When enabled, it takes the oscillator start-up time before STATUSA[OSC_RDY] becomes 1, and the clock is output to other modules.

23.3.2 Low-power modes

VBAT remains functional in all low-power modes.

23.3.3 Debug mode

Debug modes do not affect VBAT.

23.3.4 Clocks

VBAT implements the following clock domains:

- Bus interface clock - access the registers
- FRO16K internal clock
- Crystal oscillator 32.768 kHz clock

23.3.5 Reset

Only the POR resets VBAT.

23.3.6 Interrupts

VBAT generates one interrupt combining the following sources:

- VBAT POR
- Crystal oscillator ready

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

23.4 External signals

Table 145. External signals

Signal	Description	Direction
EXTAL32K	Oscillator input for 32.768 kHz crystal.	Input
XTAL32K	Oscillator output for 32.768 kHz crystal.	Output

23.5 Initialization

To enable and lock the FRO16K:

1. Write 1h to FROCTLA[FRO_EN].
2. Write 1h to FROLCKA[LOCK].
3. Alter FROCLKE[CLKE] to clock gate different FRO16K outputs to different peripherals to reduce power consumption.

To configure and lock OSC32kHz:

1. Configure OSCCTLA[EXTAL_CAP_SEL], OSCCTLA[XTAL_CAP_SEL] and OSCCTLA[COARSE_AMP_GAIN] as required based on the external crystal component ESR and CL values, and by the PCB parasitics on the EXTAL32K and XTAL32K pins. Configure 1h to OSCCTLA[CAP_SEL_EN], and 1h to OSCCTLA[OSC_EN].
2. Wait forSTATUSA[OSC_RDY] to become 1.
3. Write 1h to OSCLCKA[LOCK].
4. Alter OSCCLKE[CLKE] to clock gate different OSC32K outputs to different peripherals to reduce power consumption.

23.6 Memory map and register definition

This section includes VBAT memory map and detailed descriptions of all registers.

NOTE

The VBAT registers are reset on VBAT Cold Reset only (VBAT POR).

23.6.1 VBAT register descriptions

23.6.1.1 VBAT memory map

VBAT0 base address: 4009_3000h

Offset	Register	Width (In bits)	Access	Reset value
0h	Version ID (VERID)	32	R	0101_0007h
10h	Status A (STATUSA)	32	RW	0000_0001h
18h	Interrupt Enable A (IRQENA)	32	RW	0000_0000h
100h	Oscillator Control A (OSCCTLA)	32	RW	0000_0000h
108h	Oscillator Configuration A (OSCCFGA)	32	RW	0000_0000h
118h	Oscillator Lock A (OSCLCKA)	32	RW	0000_0000h

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Offset	Register	Width (In bits)	Access	Reset value
120h	Oscillator Clock Enable (OSCCLKE)	32	RW	0000_0000h
200h	FRO16K Control A (FROCTLA)	32	RW	0000_0001h
218h	FRO16K Lock A (FROLCKA)	32	RW	0000_0000h
220h	FRO16K Clock Enable (FROCLKE)	32	RW	0000_0000h
700h	Wakeup 0 Register A (WAKEUP0A)	32	RW	0000_0000h
708h	Wakeup 1 Register A (WAKEUP1A)	32	RW	0000_0000h
7F8h	Wakeup Lock A (WAKLCKA)	32	RW	0000_0000h

23.6.1.2 Version ID (VERID)

Offset

Register	Offset
VERID	0h

Function

Records the specific version of VBAT in the chip.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	MAJOR								MINOR							
W																
Reset	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	FEATURE															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1

Fields

Field	Function
31-24 MAJOR	Major Version Number Returns the major version number for the specification.

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
23-16 MINOR	Minor Version Number Returns the minor version number for the specification.
15-0 FEATURE	Feature Specification Number Returns the feature set number, indicating the feature set present in this instance of VBAT.

23.6.1.3 Status A (STATUSA)

Offset

Register	Offset
STATUSA	10h

Function

Contains all status flags for VBAT.

See the chip's security reference manual for security-related field information.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0												0	0	0	0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0	0	0	OSC_RDY	0				POR_DET
W																W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Fields

Field	Function
31-20 —	Reserved
19 —	Reserved

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
18 —	Reserved
17 —	Reserved
16 —	Reserved
15 —	Reserved
14 —	Reserved
13 —	Reserved
12 —	Reserved
11 —	Reserved
10 —	Reserved
9 —	Reserved
8 —	Reserved
7 —	Reserved
6 —	Reserved
5 OSC_RDY	OSC32k Ready Indicates whether OSC32k is enabled and the clock output is stable. 0b - Disabled (clock not ready)

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	1b - Enabled (clock ready)
4-1 —	Reserved
0 POR_DET	POR Detect Flag Indicates if the VBAT POR has asserted. **NOTE** This field behaves differently for register reads and writes. When reading 0b - Not reset 1b - Reset When writing 0b - No effect 1b - Clear the flag

23.6.1.4 Interrupt Enable A (IRQENA)

Offset

Register	Offset
IRQENA	18h

Function

Contains all interrupt enables for VBAT.

NOTE

See the chip's security reference manual for security-related field information.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0												0	0	0	0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0	0	0	OSC_RDY	0				POR_DET
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Fields

Field	Function
31-20 —	Reserved
19 —	Reserved
18 —	Reserved
17 —	Reserved
16 —	Reserved
15 —	Reserved
14 —	Reserved
13 —	Reserved
12 —	Reserved
11 —	Reserved
10 —	Reserved
9 —	Reserved
8 —	Reserved
7 —	Reserved
6	Reserved

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
—	
5 OSC_RDY	OSC32k Ready Enables the corresponding interrupt in Status A (STATUSA). 0b - Disable 1b - Enable
4-1 —	Reserved
0 POR_DET	POR Detect Enables the corresponding interrupt in Status A (STATUSA). 0b - Disable 1b - Enable

23.6.1.5 Oscillator Control A (OSCCTLA)

Offset

Register	Offset
OSCCTLA	100h

Function

Contains the oscillator control fields. Writes to this register are blocked when OSCLCKA[LOCK] is 1.

Some fields in this register uses the following equation to calculate the load capacitance:

$$C_L = \frac{C_{extal} \times C_{xtal}}{C_{extal} + C_{xtal}} + C_{shunt}$$

Equation 1. Equation for load capacitance

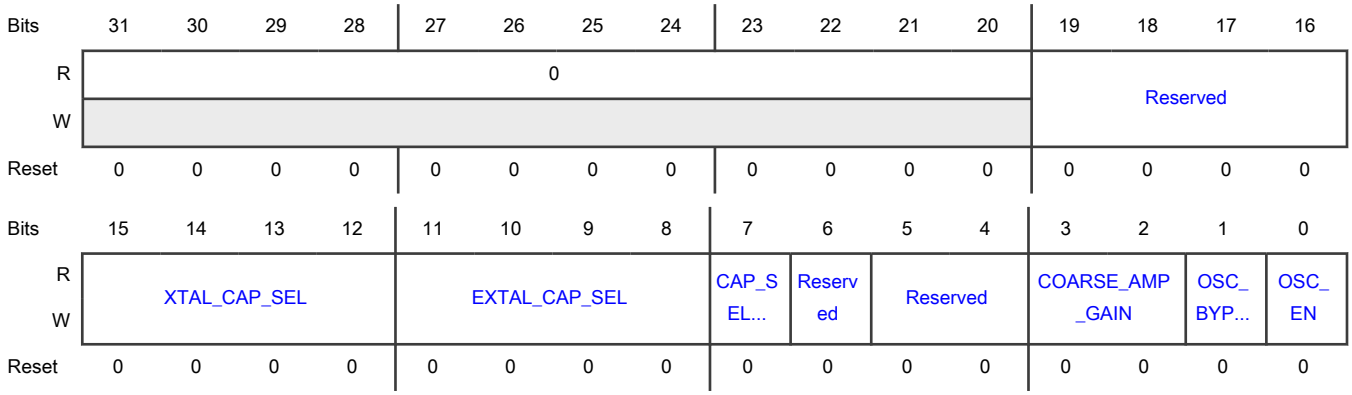
where:

- C_{extal} is the selected capacitance on the EXTAL pin.
- C_{xtal} is the selected capacitance on the XTAL pin.
- C_{shunt} is the internal shunt capacitance. You can determine this capacitance value from the chip's data sheet.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram



Fields

Field	Function
31-20 —	Reserved
19-16 —	Reserved
15-12 XTAL_CAP_SEL	Crystal Load Capacitance Selection Selects the internal capacitance for the XTAL pin from the capacitor bank. Calculate the final load capacitance as shown in Equation 1. You must write 0000b to this field in low power mode. 0000b - 0 pF 0001b - 2 pF 0010b - 4 pF 0011b - 6 pF 0100b - 8 pF 0101b - 10 pF 0110b - 12 pF 0111b - 14 pF 1000b - 16 pF 1001b - 18 pF 1010b - 20 pF 1011b - 22 pF 1100b - 24 pF 1101b - 26 pF

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	1110b - 28 pF 1111b - 30 pF
11-8 EXTAL_CAP_SEL	Crystal Load Capacitance Selection Selects the internal capacitance for the EXTAL pin from the capacitor bank. Calculate the final load capacitance as shown in Equation 1. The configuration EXTAL_CAP_SEL=0000 and CAP_SEL_EN=1 is required in low power mode and is not supported in other modes 0000b - 0 pF 0001b - 2 pF 0010b - 4 pF 0011b - 6 pF 0100b - 8 pF 0101b - 10 pF 0110b - 12 pF 0111b - 14 pF 1000b - 16 pF 1001b - 18 pF 1010b - 20 pF 1011b - 22 pF 1100b - 24 pF 1101b - 26 pF 1110b - 28 pF 1111b - 30 pF
7 CAP_SEL_EN	Crystal Load Capacitance Selection Enable Enables the internal capacitor banks on the EXTAL and XTAL pins that vary the load capacitance on these pins. You must write 1 to this field and OSCCTLA[OSC_EN] simultaneously. **NOTE** The configuration EXTAL_CAP_SEL=0000 and CAP_SEL_EN=1 is required in low power mode and is not supported in other modes 0b - Disable 1b - Enable
6 —	Reserved

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
5-4 —	Reserved
3-2 COARSE_AMP_GAIN	Amplifier Gain Coarse Adjustment Allows the use of a wide range of external crystal ESR values. Tunes the internal transconductance (g_m) by increasing the current. **NOTE** See the device datasheet for the ranges supported by this device 00b - ESR Range 0 01b - ESR Range 1 10b - ESR Range 2 11b - ESR Range 3
1 OSC_BYP_EN	Crystal Oscillator Bypass Enable Bypasses the crystal oscillator. The oscillator outputs the clock, but this clock is the same as the clock provided on EXTAL pin. You must write 1 to both OSCCTLA[OSC_EN] and this field to enable the Bypass mode of the oscillator. To exit Bypass mode, you must write 0 to both OSCCTLA[OSC_EN] and this field. 0b - Does not bypass 1b - Bypass
0 OSC_EN	Crystal Oscillator Enable Enables the crystal oscillator. The oscillator starts giving the clock output but the clock is not available to the chip until STATUSA[OSC_RDY] is asserted. 0b - Disable 1b - Enable

23.6.1.6 Oscillator Configuration A (OSCCFGA)

Offset

Register	Offset
OSCCFGA	108h

Function

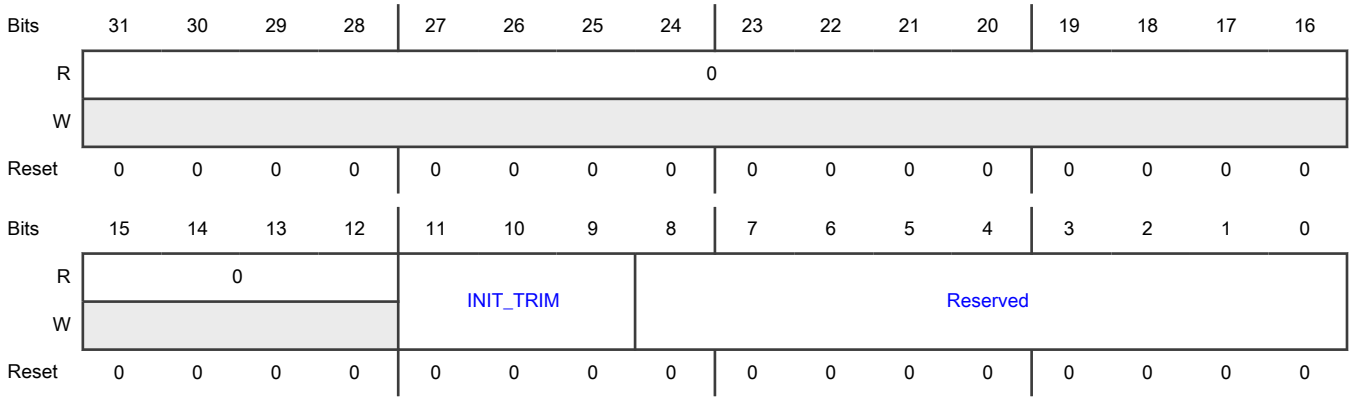
Contains the oscillator configuration fields. Writes to this register are blocked when OSCLCKA[LOCK] is 1.

Reset values are loaded out of IFR.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram



Fields

Field	Function
31-12 —	Reserved
11-9 INIT_TRIM	Initialization Trim Configures the start-up time of the oscillator. 000b - 8 s 001b - 4 s 010b - 2 s 011b - 1 s 100b - 0.5 s 101b - 0.25 s 110b - 0.125 s 111b - 0.5 ms
8-0 —	Reserved

23.6.1.7 Oscillator Lock A (OSCLCKA)

Offset

Register	Offset
OSCLCKA	118h

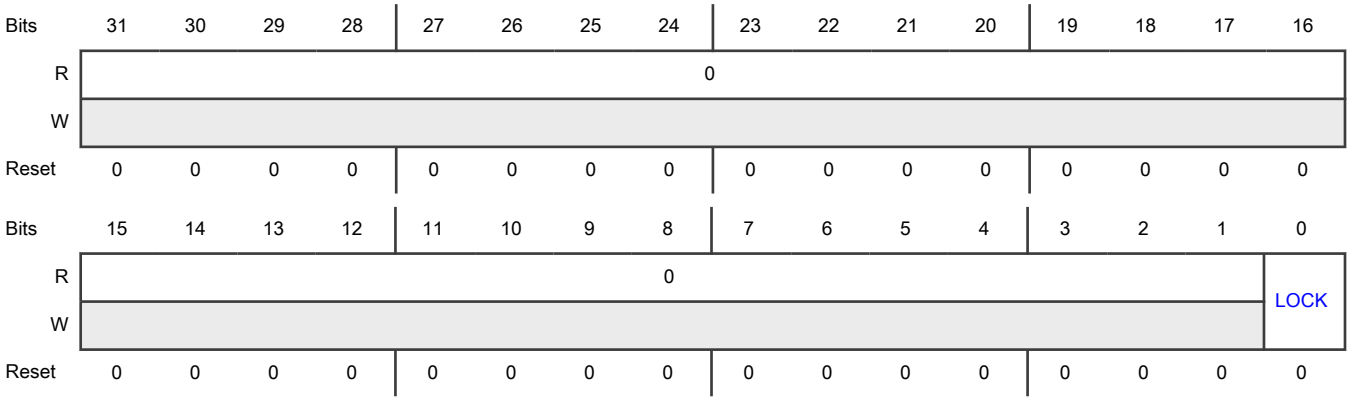
Function

Allows you to block any write to the oscillator registers. Writes to this register are blocked when OSCLCKA[LOCK] is 1.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram



Fields

Field	Function
31-1 —	Reserved
0 LOCK	Lock Blocks any write to the oscillator registers. 0b - Do not block 1b - Block

23.6.1.8 Oscillator Clock Enable (OSCCLKE)

Offset

Register	Offset
OSCCLKE	120h

Function

Contains clock gating register field to gate the OSC32k clock to other modules.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0														CLKE	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-2 —	Reserved
1-0 CLKE	Clock Enable Enables the corresponding OSC32 kHz output clock to other modules when you write 1 to this field. See the chip-specific VBAT information section for more information.

23.6.1.9 FRO16K Control A (FROCTLA)

Offset

Register	Offset
FROCTLA	200h

Function

Controls FRO16K. Writes to this register are blocked when FROLCKA[LOCK] is 1.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															FRO_EN
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Fields

Field	Function
31-1 —	Reserved
0 FRO_EN	FRO16K Enable 0b - Disable 1b - Enable

23.6.1.10 FRO16K Lock A (FROLCKA)

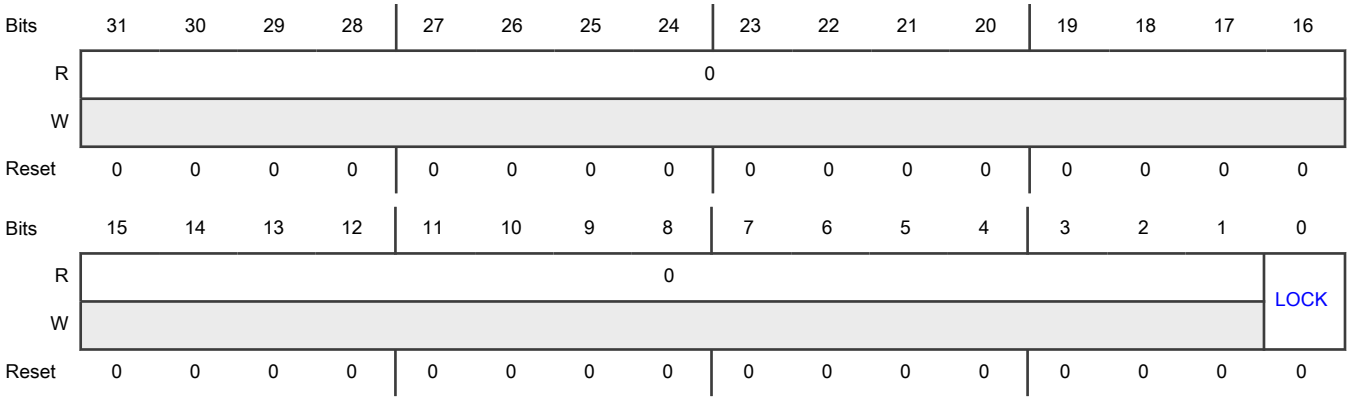
Offset

Register	Offset
FROLCKA	218h

Function

Contains the lock field. Writes to this register are blocked when FROLCKA[LOCK] is 1.

Diagram



Fields

Field	Function
31-1 —	Reserved
0 LOCK	Lock Blocks any write to the FRO16K registers when you write 1 to this field. VBAT POR writes 0 to this field.

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	0b - Do not block 1b - Block

23.6.1.11 FRO16K Clock Enable (FROCLKE)

Offset

Register	Offset
FROCLKE	220h

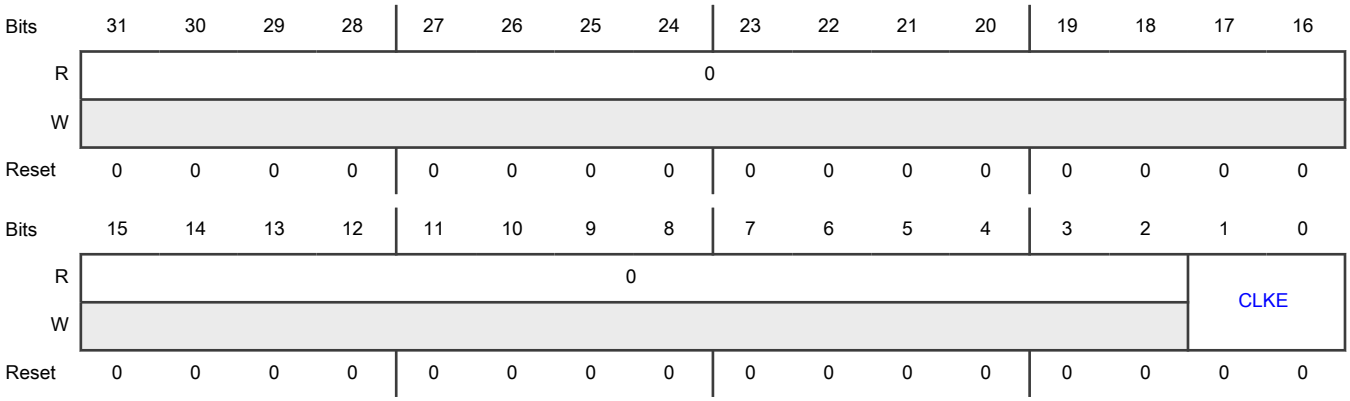
Function

Contains clock gating register field to gate the FRO16K clock to other modules.

NOTE

FROCLKE cannot be locked (not affected by FRO16K Lock A (FROLCKA)).

Diagram



Fields

Field	Function
31-2 —	Reserved
1-0 CLKE	Clock Enable Enables the corresponding FRO16 kHz output clock to other modules when you write 1 to the corresponding bit in this field. See the chip-specific VBAT information section for more information.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

23.6.1.12 Wakeup a Register A (WAKEUP0A - WAKEUP1A)

Offset

Register	Offset
WAKEUP0A	700h
WAKEUP1A	708h

Function
Contains software writable bits. Writes to this register are blocked when WAKEUP lock register is set.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	REG															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	REG															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-0 REG	Register Software writable value.

23.6.1.13 Wakeup Lock A (WAKLCKA)

Offset

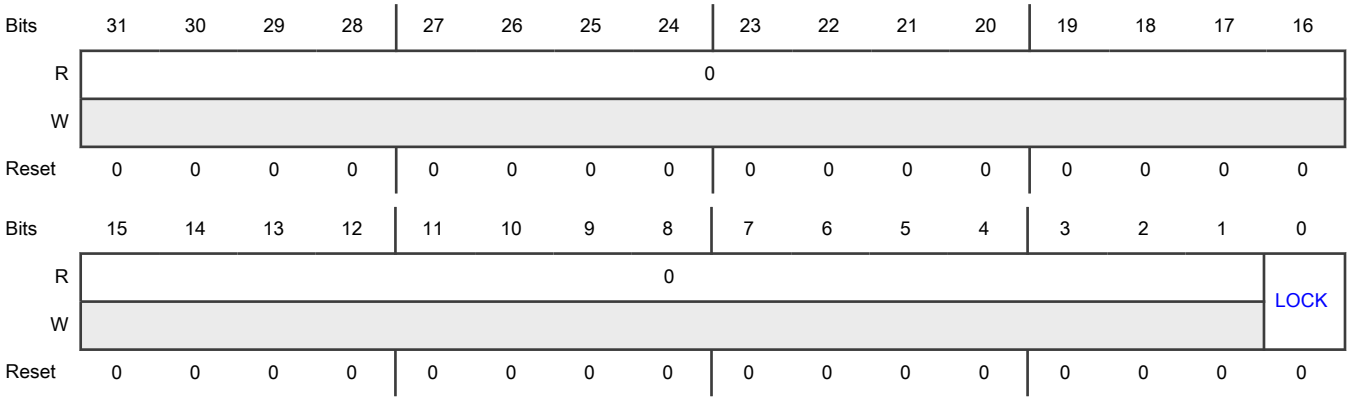
Register	Offset
WAKLCKA	7F8h

Function
Contains the lock bits. Writes to this register are blocked when Wakeup lock registers are set.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram



Fields

Field	Function
31-1 —	Reserved
0 LOCK	Lock When set, blocks writes to the Wakeup registers. 0b - Lock is disabled 1b - Lock is enabled

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Chapter 24
System Power Control (SPC)

24.1 Chip-specific SPC information

Table 146. Reference links to related information

Topic	Related module	Reference
Full description	SPC	SPC
System memory map		Memory map
Clocking		Clock distribution
Power Management		Power Management
Signal multiplexing	Port control	Signal multiplexing

24.1.1 Module instances

This chip has one instance of the SPC module, SPC0.

24.1.2 Voltage regulators

This section describes the various on-chip regulators.

Table 147. On-chip regulators

Regulator name	Description
LDO_CORE	The LDO_CORE can be used to power VDD_CORE domain.
RAM_RET_LDO	Supports only one SRAM retention switch, which supports up to 16 KB SRAM. • SRAM_RET_EN[0]: RAM A0/A1 , • SRAM_RET_EN[3:1]: Reserved

24.1.3 SOC_CNTRL fields

ACTIVE_CFG1[SOC_CNTRL] field controls which analog modules are enabled and disabled in Active or Sleep modes. LP_CFG1[SOC_CNTRL] field controls which analog modules are enabled and disabled in other low-power modes (like Deep Sleep, Power Down, or Deep Power Down). The following table lists the analog modules that these fields control.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R/W												CMP0_DAC_EN				CMP0_EN
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

All information provided in this document is subject to legal disclaimers.
© 2026 NXP B.V. All rights reserved.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W								OPAMP0_EN								
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

24.1.4 Low power current reference enable bit LP_CFG[LP_IREFEN]

The low power current reference is used by analog modules CMP0, and FRO12M. The bit LP_IREFEN should be set if these analog modules are used in the Deep Power Down mode. For CMP0, this bit should be set if using nano mode in the Deep Power Down mode.

24.1.5 Low power wake-up delay

Table 148. LPWKUP delay

Regulator	LP_CFG voltage	ACTIVE_CFG voltage	Minimum delay	Minimum LPWKUP_DELAY
LDO_CORE	0.6 V	1.1 V	12 µs	6Eh

24.2 Overview

SPC contains and controls the following components:

- Regulator for SRAM domain providing the retention voltage to SRAM array in low power modes.
- One low-drop-out (LDO) voltage regulator (1.1 V LDO_CORE)
- Circuits that monitor and assure correct voltage levels:
  - Power-on reset (POR)
  - Low-voltage detect (LVD)
  - High-voltage detect (HVD)

SPC turns these components on and off, and switches them to different power mode levels based on power-mode change requests from each chip power domain.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

24.2.1 Block diagram

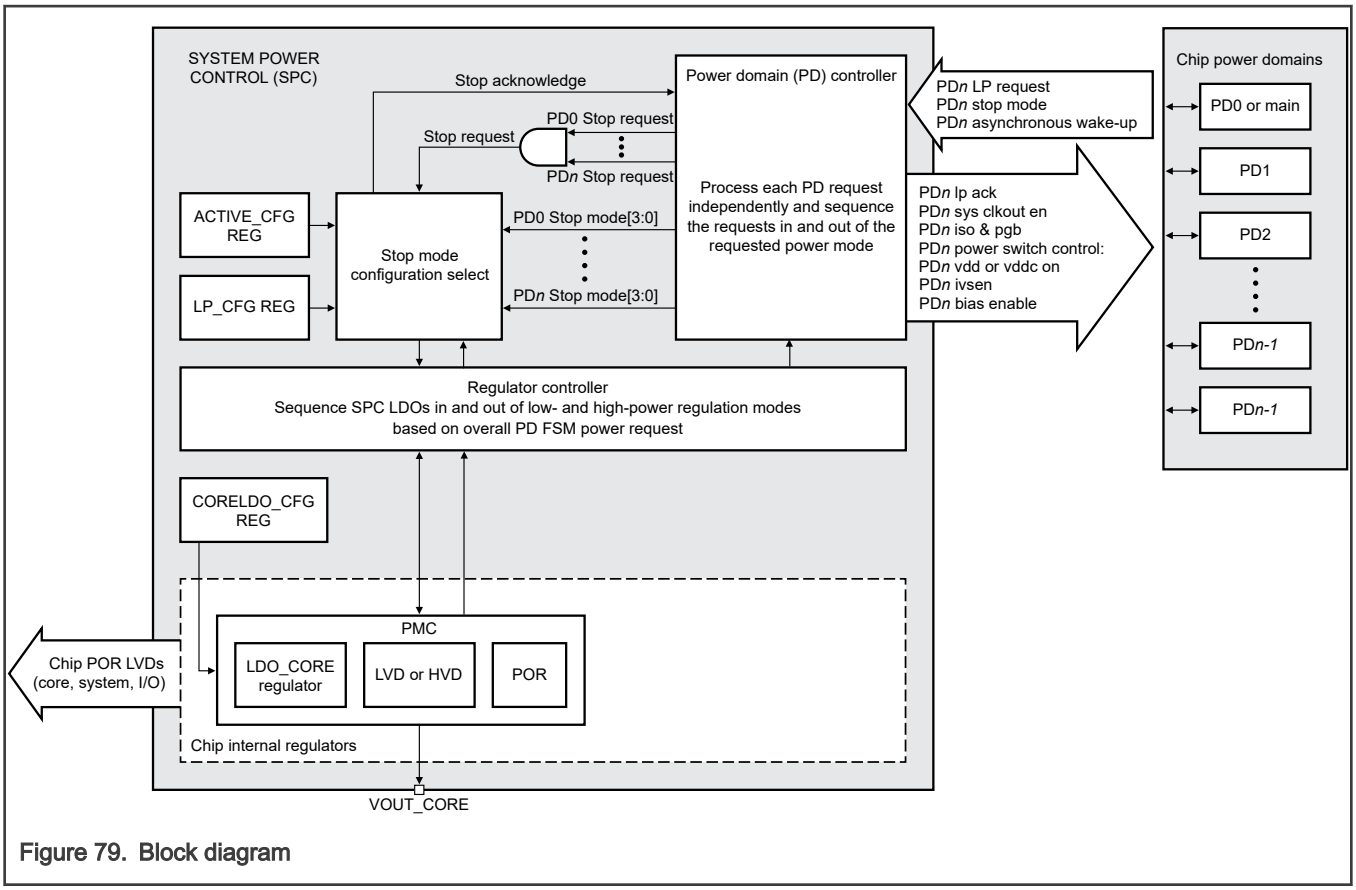


Figure 79. Block diagram

24.2.2 Features

- Selectable core internal voltage regulator (LDO_CORE)
- Active POR providing brown-out detect
- Low-voltage detect
- High-voltage detect
- Core power gate control

24.3 Functional description

24.3.1 Power mode transitions

Figure 80 shows the power mode state transitions available for each power-down domain. A POR or LVD reset initially returns the power-down domains to ACTIVE state.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

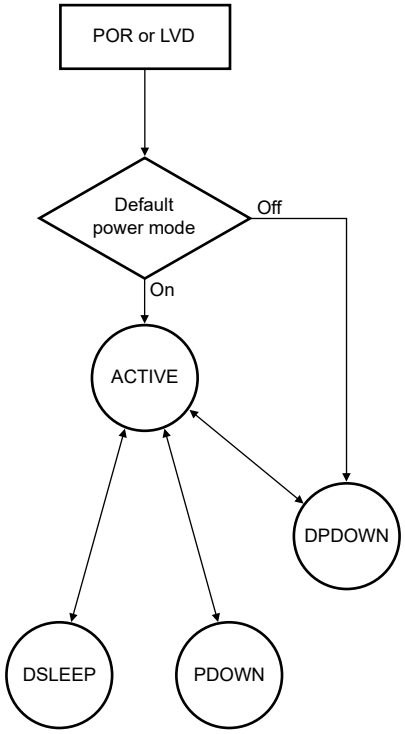


Figure 80. Power mode state diagram

SPC allows each power domain to enter or exit low-power modes independently of the other. You must ensure that any common system resources (for example, clock sources) are always available whenever required for a power-domain current power mode. For example, if a clock source in power domain 0 drives power domain 1, you must ensure that power domain 0 does not enter a low-power mode, because that would disable the clock source.

Figure 81 defines the low-power entry and exit flow for each of the SPC power-down domains.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

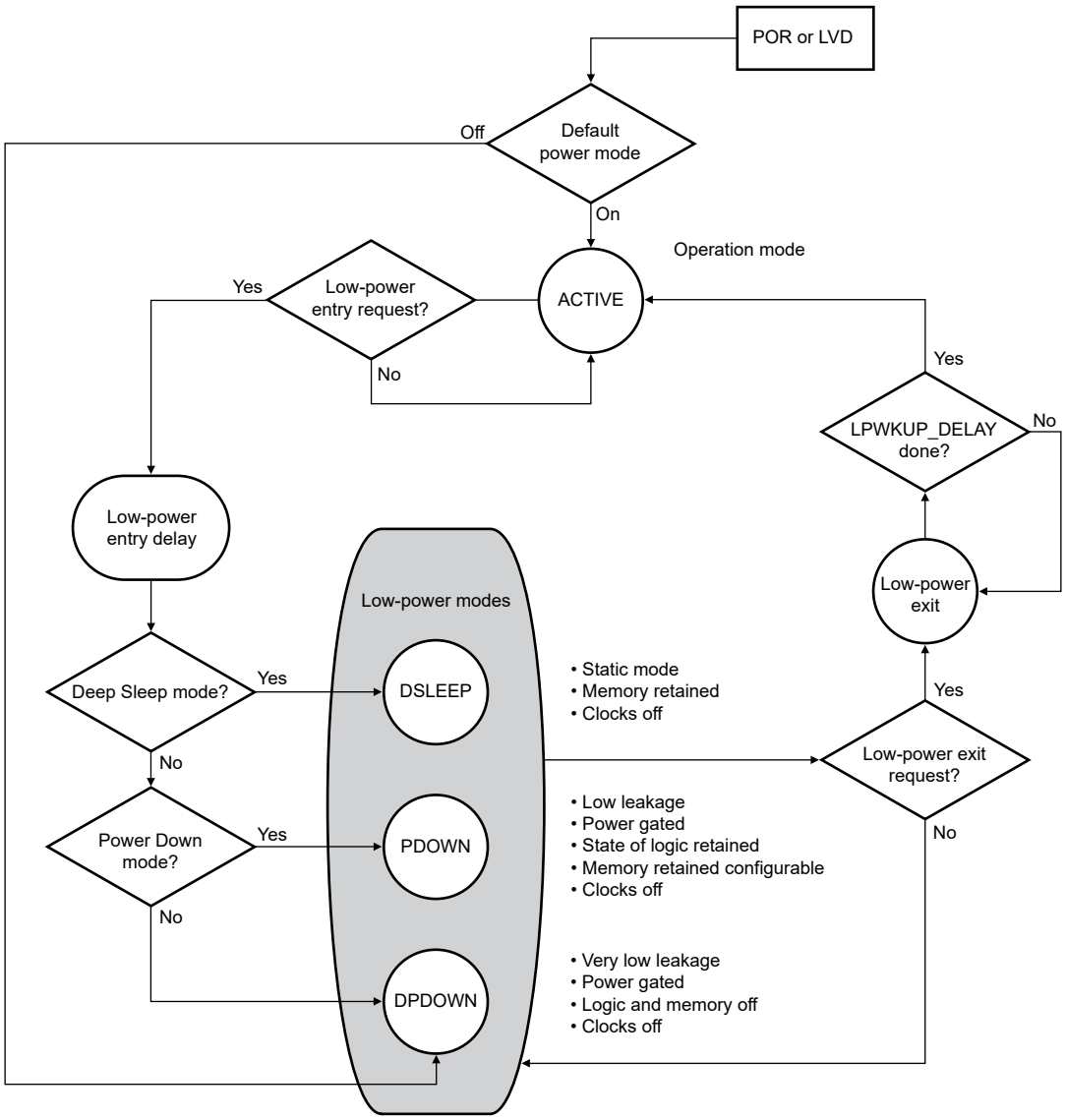


Figure 81. SPC regulator power mode state diagram

NOTE

This section mentions the power modes supported by the module. For the power modes supported on your device, see the chip-specific Power management chapter.

24.3.2 Active mode voltage updates

This section describes voltage and frequency changes in Active mode.

You can alter the operating voltage and frequency during Active mode to optimize power consumption depending on the application's performance requirements.

When increasing voltage and frequency in Active mode, you must perform the following steps:

1. Increase voltage to a new level (ACTIVE_CFG[CORELDO_VDD_LVL]).
2. Wait for voltage change to complete (SC[BUSY] = 0).
3. Configure flash memory to support higher voltage level and frequency (FMU_FCTRL[RWSC].

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

4. Configure SRAM to support higher voltage levels (SRAMCTL[VSM]).
5. Request SRAM voltage update (write 1 to SRAMCTL[REQ]).
6. Wait for SRAM voltage change to complete (SRAMCTL[ACK] = 1).
7. Clear request for SRAM voltage change (write 0 to SRAMCTL[REQ]).
8. Increase frequency to a new level (for example, SCG_RCCR).
9. You can continue execution.

When decreasing voltage and frequency in Active mode perform the following steps:

1. Decrease frequency to a new level (for example, SCG_RCCR).
2. Configure flash memory to support lower voltage level and frequency (FMU_FCTRL[RWSC].
3. Configure SRAM to support lower voltage levels (SRAMCTL[VSM]).
4. Request SRAM voltage update (write 1 to SRAMCTL[REQ]).
5. Wait for SRAM voltage change to complete (SRAMCTL[ACK] = 1).
6. Clear request for SRAM voltage change (write 0 to SRAMCTL[REQ]).
7. Decrease voltage to a new level).
8. Wait for voltage change to complete (SC[BUSY] = 0).
9. You can continue execution.

24.3.3 Low-Power Request (LPREQ) pin

The LPREQ pin asserts after low-power entry and negates after low-power wakeup.

The pin is intended to communicate with an external PMIC (for example, to enter low-power mode) or to control external switches (for example, if certain power supply rails are switched off in the low-power mode). The pin can eliminate the latency when communicating with PMIC—for example, through I^2C—for quick power-state transitions. The PMIC can be configured to use the LPREQ pin to switch between two preconfigured power states.

SPC controls the state of the LPREQ pin based on how you configure Low-Power Request Configuration (LPREQ_CFG). You control the LPREQ pin in Active mode. SPC controls the pin when the chip transitions from Active to a low-power mode, and after wake-up from these power modes.

NOTE

If the power supply rails are switched off externally, software must configure the internal isolation of these power domains using External Voltage Domain Configuration (EVD_CFG).

To use the LPREQ pin:

1. Specify the pin polarity (LPREQ_CFG[LPREQPOL]).
2. Enable the pin output (LPREQ_CFG[LPREQOE]).
3. Configure the pin mux for the desired pin using the PORT PCR registers.
4. If the external PMIC or switch needs additional time after wake-up, use Low Power Wake-Up Delay (LPWKUP_DELAY) to extend the wake-up time.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

24.3.4 Clocking

Table 149. Clocks

SPC component:	Clock source
Registers and logic	Slow bus clock
Delay counters	10 MHz internal clock

SPC uses the internal clock that PMC analog block generates to sequence the system regulator and chip power domains in and out of different power modes. This clock has a frequency of 10 MHz. SPC automatically enables the clock when it detects a power mode change, including entry into low-power modes and exit from low-power modes of any of the system power domains, or configuration changes to either Active Power Mode Configuration (ACTIVE_CFG) or Low-Power Mode Configuration (LP_CFG). This clock cannot run unless the PMC bandgap is up and enabled. Once SPC completes all transactions and is not busy, that is, SC[BUSY] = 0, this clock will turn off synchronously.

24.3.5 Reset

24.3.5.1 Reset sources

24.3.5.1.1 Power-on reset (POR)

When you initially apply power to the chip, or the supply voltage is below the POR falling threshold, the POR circuit triggers the POR condition. The POR condition asserts a cold reset in all power domains.

24.3.5.1.2 Low-voltage detect (LVD)

SPC supports the following LVD circuits:

- Core VDD
- System VDD

Each of these circuits is enabled by default and keeps the chip in reset until each of these supply voltages rises above the LVD rising threshold. Enabling any of the LVD circuits triggers an LVD reset condition if the corresponding supply voltage is below the LVD falling threshold.

The LVD reset condition asserts a cold reset in all power domains. The LVD- and HVD-detection logic is the only logic that HVD and LVD resets do not affect. The corresponding status fields in Voltage Detect Status (VD_STAT) become 1 after the corresponding LVD reset condition occurs.

24.3.5.1.3 High-voltage detect (HVD)

SPC supports the following HVD circuit:

- System VDD

Each of these circuits is enabled by default and triggers an HVD reset condition if the corresponding supply voltage is above the HVD threshold and also the VD_CORE_CFG[HVDRE] and VD_SYS_CFG[HVDRE] for each corresponding HVD is enabled.

Any of these reset condition asserts a cold reset in all power domains. The LVD- and HVD-detection logic is the only logic that HVD and LVD resets do not affect. The corresponding status fields in Voltage Detect Status (VD_STAT) become 1 after the corresponding HVD reset condition occurs.

24.3.5.2 Reset sequence

The following steps occur as part of the POR, LVD, or HVD sequence:

1. The RESET_b pin deasserts.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

2. The internal reset signals assert.
3. The POR or LVD signals negate.
4. System clocks are enabled.

24.3.6 Interrupts

SPC generates a single interrupt. The trigger for this interrupt comes from:

- Any of the LVD or HVD circuits when they are not configured for reset and their corresponding HVDIE or LVDIE register.

24.4 External signals

Signal	Description	Direction
LPREQ	Low-power request pin used to signal external power-management circuits for a change in supply voltage	Output

24.5 Initialization

SPC does not require any special initialization.

24.6 Application information

To disable Bandgap in Active mode:

1. Disable all LVD's and HVD's in ACTIVE_CFG[29:24]= 0x00
2. Configure LDO's to Low Drive Strength in ACTIVE_CFG register
3. Configure ACTIVE_CFG[BGMODE] = 0x0

To disable Bandgap in Low-Power mode:

1. Disable all LVD's and HVD's in LP_CFG[29:24]= 0x00
2. Configure LDO's to Low Drive Strength in LP_CFG register
3. Configure LP_CFG[BGMODE] = 0x0

24.7 SPC register descriptions

Different chip reset types affect the reset of different SPC registers. Each register description provides details. See the Reset chapter for more information about the types of reset on this chip.

You may use only 32-bit writes for any writable SPC registers. 8-bit or 16-bit writes cause a transfer error.

24.7.1 SPC memory map

SPC0 base address: 4009_0000h

Offset	Register	Width (In bits)	Access	Reset value
0h	Version ID (VERID)	32	R	0000_0000h
10h	Status Control (SC)	32	RW	0000_0000h

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Offset	Register	Width (In bits)	Access	Reset value
14h	SPC Regulator Control (CNTRL)	32	RW	0000_0001h
1Ch	Low-Power Request Configuration (LPREQ_CFG)	32	RW	0000_0000h
20h	SPC Configuration (CFG)	32	RW	0000_0000h
30h	SPC Power Domain Mode Status (PD_STATUS0)	32	RW	0000_0000h
40h	SRAM Control (SRAMCTL)	32	RW	0000_0002h
58h	SRAM Retention LDO Control (SRAMRETLDO_CNTRL)	32	RW	0000_0F01h
100h	Active Power Mode Configuration (ACTIVE_CFG)	32	RW	1310_0009h
104h	Active Power Mode Configuration 1 (ACTIVE_CFG1)	32	RW	0000_0000h
108h	Low-Power Mode Configuration (LP_CFG)	32	RW	0008_1008h
10Ch	Low Power Mode Configuration 1 (LP_CFG1)	32	RW	0000_0000h
120h	Low Power Wake-Up Delay (LPWKUP_DELAY)	32	RW	0000_0000h
124h	Active Voltage Trim Delay (ACTIVE_VDELAY)	32	RW	0000_00C8h
130h	Voltage Detect Status (VD_STAT)	32	RW	0000_0000h
134h	Core Voltage Detect Configuration (VD_CORE_CFG)	32	RW	0000_0001h
138h	System Voltage Detect Configuration (VD_SYS_CFG)	32	RW	0000_0101h
140h	External Voltage Domain Configuration (EVD_CFG)	32	RW	0000_0000h
300h	LDO_CORE Configuration (CORELDO_CFG)	32	RW	0000_0000h

24.7.2 Version ID (VERID)

Offset

Register	Offset
VERID	0h

Function

Indicates:

- The version integrated for this instance on the chip.
- Inclusion or exclusion of several optional features.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	MAJOR								MINOR							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	FEATURE															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-24 MAJOR	Major Version Number
23-16 MINOR	Minor Version Number
15-0 FEATURE	Feature Specification Number Indicates the feature set number. 0000_0000_0000_0000b - Standard features All other values are reserved.

24.7.3 Status Control (SC)

Offset

Register	Offset
SC	10h

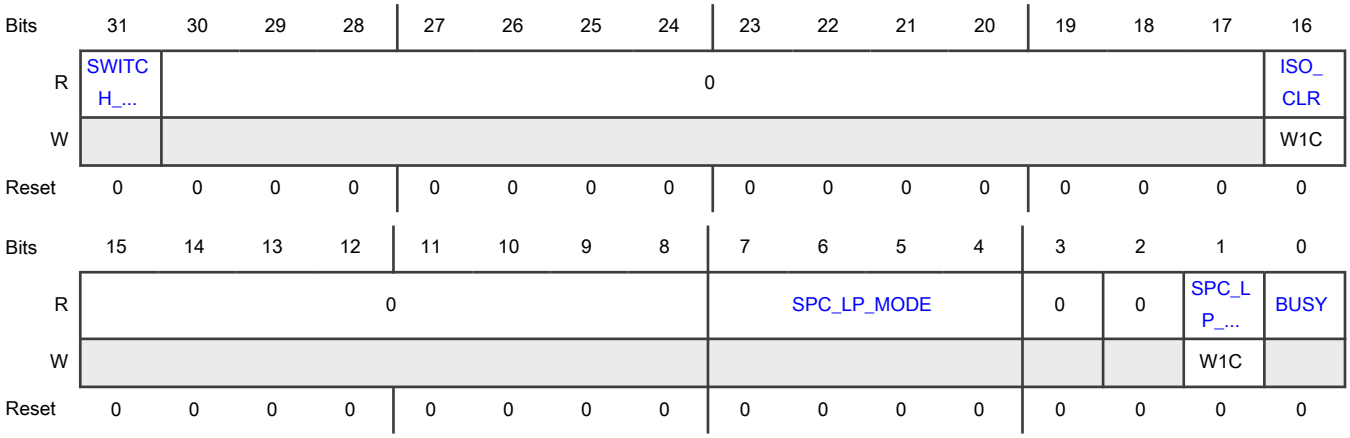
Function

Indicates the current SPC status.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram



Fields

Field	Function
31 SWITCH_STATE	Power Switch State Indicates the power switch state. 0b - Off 1b - On
30-17 —	Reserved
16 ISO_CLR	Isolation Clear Flags Indicates whether certain peripherals and I/O pads are in a latched state as a result of having been in DPDOWN mode. Each bit of this field is a flag associated with a module on this chip. See the chip-specific SPC information for the module-bit association. For each flag: • A value of 0 means that peripherals and I/O pads are in the normal run state. • A value of 1 means that peripherals and I/O pads can retain their state in their power domain. If a flag in this field is 1, writing 1 to that flag clears the flag and releases the I/O pads and peripherals to their normal run-mode state. After recovering from a power-down mode, you must restore the chip configuration before clearing a flag in this field. In particular, you must restore the pin configuration for enabled WUU wake-up to avoid any WUU flag from being falsely set when the associated ISO_CLR flag is cleared. This field resets after a system reset(not Deep Power Down wakeup reset).
15-8 —	Reserved
7-4	Power Domain Low-Power Mode Request Indicates the last low-power mode that the power domain requested.

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
SPC_LP_MODE	0000b - Sleep mode with system clock running 0001b - DSLEEP with system clock off 0010b - PDOWN with system clock off 0100b - Reserved 1000b - DPDOWN with system clock off
3 —	Reserved
2 —	Reserved
1 SPC_LP_REQ	SPC Power Mode Configuration Status Flag Indicates when all power-down domains requested low-power mode and SPC entered a low-power state. **NOTE** This field behaves differently for register reads and writes. When reading 0b - SPC is in Active mode; the ACTIVE_CFG register has control 1b - All power domains requested low-power mode; SPC entered a low-power state; power-mode configuration based on the LP_CFG register When writing 0b - No effect 1b - Clear the flag
0 BUSY	SPC Busy Status Flag Indicates whether SPC is busy. SPC sets this flag: • When SPC executes any type of power mode transition in Active mode or any of the chip low-power modes. • Due to LP_CFG[CORELDO_VDD_LVL] changes while in Active mode. **NOTE** Wait until this flag is clear before changing power-mode configuration registers. 0b - Not busy 1b - Busy

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

24.7.4 SPC Regulator Control (CNTRL)

Offset

Register	Offset
CNTRL	14h

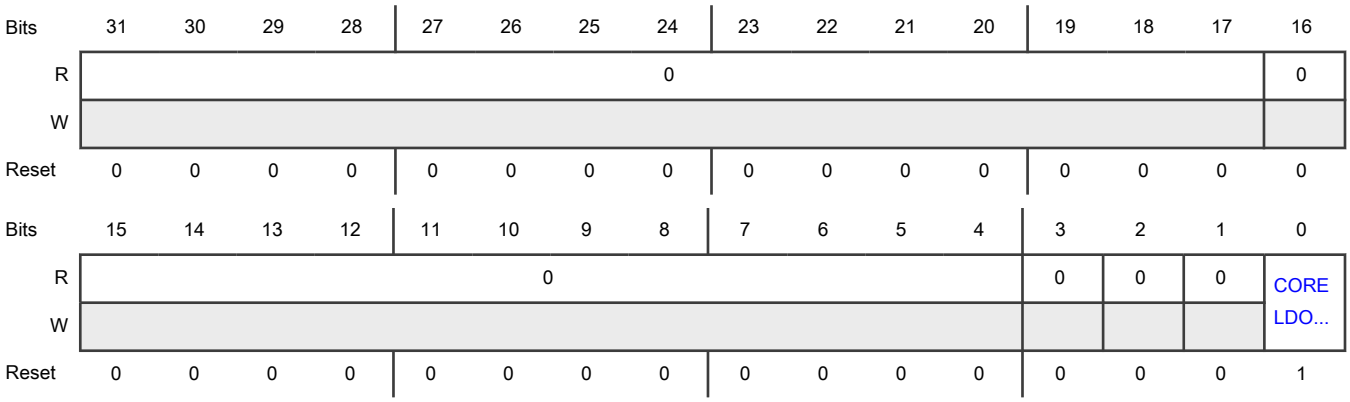
Function

Enables the SPC regulators.

You can write to the fields in this register only one time.

The register resets only after a POR, LVD, or HVD event.

Diagram



Fields

Field	Function
31-17 —	Reserved
16 —	Reserved
15-4 —	Reserved
3 —	Reserved
2 —	Reserved
1	Reserved

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
—	
0 CORELDO_EN	LDO_CORE Regulator Enable 0b - Disable 1b - Enable

24.7.5 Low-Power Request Configuration (LPREQ_CFG)

Offset

Register	Offset
LPREQ_CFG	1Ch

Function

Configures the low-power output request pin.

The register resets only after a POR, LVD, or HVD event.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0								0							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0												LPREQOV		LPREQPOL	LPREQOE
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-26 —	Reserved
25-4 —	Reserved
3-2	Low-Power Request Output Override

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
LPREQOV	Forces the low-power request pin high. 00b - Not forced 01b - Reserved 10b - Forced low (ignore LPREQPOL settings) 11b - Forced high (ignore LPREQPOL settings)
1 LPREQPOL	Low-Power Request Output Pin Polarity Control Controls the true polarity of the low-power request output pin. 0b - High 1b - Low
0 LPREQOE	Low-Power Request Output Enable Enables the low-power request output pin. 0b - Disable 1b - Enable

24.7.6 SPC Configuration (CFG)

Offset

Register	Offset
CFG	20h

Function

Controls the features of the integrated power switch that you can use to gate board-level supplies.

The register resets only after a POR, LVD, or HVD event.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0												INTG_PW...	INTG_PW...	INTG_PW...	INTG_PW...
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Fields

Field	Function
31-4 —	Reserved
3 INTG_PWSWTCH_WKUP_ACTIVE_EN	Integrated Power Switch Wake-up Enable Enables the switch in active mode. When this field is 1, SPC drives a pin in active mode to close the switch. 0b - Disable 1b - Enable
2 INTG_PWSWTCH_SLEEP_ACTIVE_EN	Integrated Power Switch Active Enable Disables the switch in active mode. When this field is 1, SPC drives a pin in active mode to open the switch. 0b - Disable 1b - Enable
1 INTG_PWSWTCH_WKUP_EN	Integrated Power Switch Wake-up Enable Enables the switch after waking up from any low-power mode. When this field is 1, SPC drives a pin during low-power wake-up sequence to close the switch. 0b - Disable 1b - Enable
0 INTG_PWSWTCH_SLEEP_EN	Integrated Power Switch Sleep Enable Disables the switch entering in any low-power mode. When this field is 1, SPC drives a pin during low-power entry sequence to open the switch. 0b - Disable 1b - Enable

24.7.7 SPC Power Domain Mode Status (PD_STATUS0)

Offset

Register	Offset
PD_STATUS0	30h

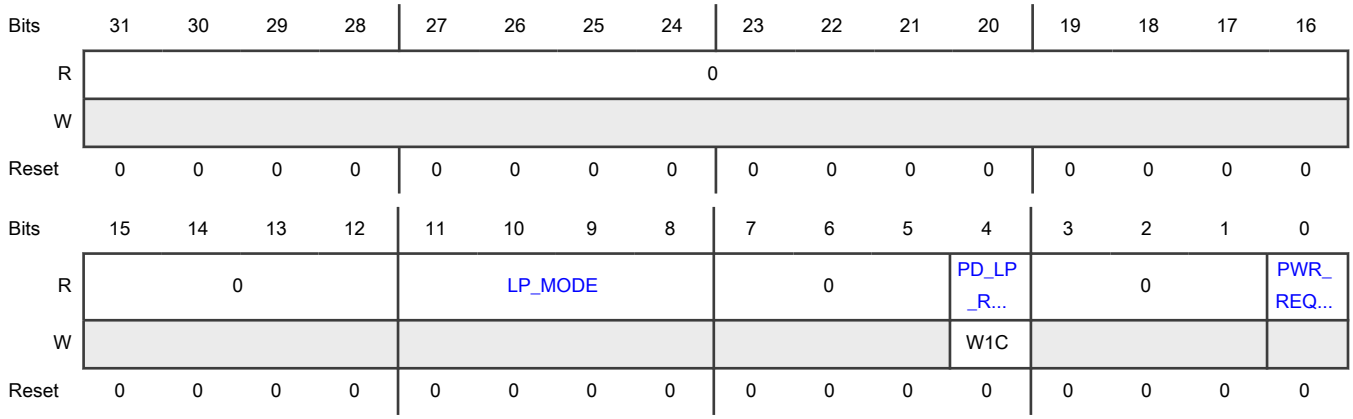
Function

Indicates power mode status for each of the SPC power domains. See the chip-specific SPC information to determine which power domains are associated with this register.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram



Fields

Field	Function
31-12 —	Reserved
11-8 LP_MODE	Power Domain Low Power Mode Request Indicates the last low-power mode that the power domain requested. 0000b - SLEEP with system clock running 0001b - DSLEEP with system clock off 0010b - PDOWN with system clock off 0100b - Reserved 1000b - DPDOWN with system clock off
7-5 —	Reserved
4 PD_LP_REQ	Power Domain Low Power Request Flag Set when low power mode was requested by Power Domain since last cleared by software. 0b - Did not request 1b - Requested
3-1 —	Reserved
0 PWR_REQ_STATUS	Power Request Status Flag Indicates whether the power domain requested low-power mode. 0b - Did not request 1b - Requested

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

24.7.8 SRAM Control (SRAMCTL)

Offset

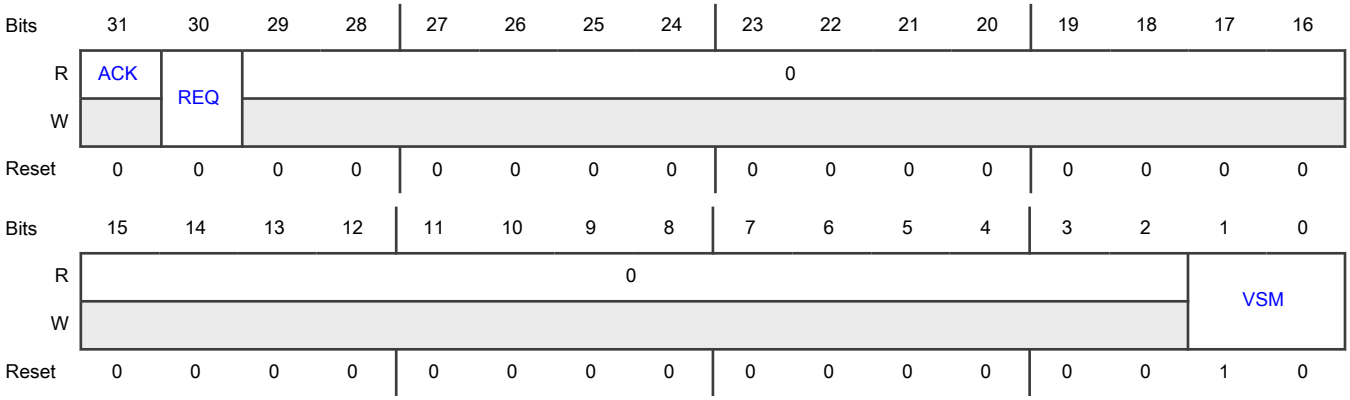
Register	Offset
SRAMCTL	40h

Function

Configures the SRAM timing for different voltage levels.

When transitioning between two voltage levels, you must keep the SRAM timing at the lowest voltage level until the voltage change is complete.

Diagram



Fields

Field	Function
31 ACK	SRAM Voltage Update Request Acknowledge Indicates whether SPC acknowledged the request for an SRAM trim-value change. 0b - Not acknowledged 1b - Acknowledged
30 REQ	SRAM Voltage Update Request Allows you to request an SRAM trim value change. After requesting the change, you must wait for the ACK field to become 1, then write 0 to REQ. 0b - Do not request 1b - Request
29-2 —	Reserved
1-0	Voltage Select Margin

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
VSM	Specifies the operating voltage for the SRAM's read/write timing margin. 00b - Reserved 01b - Reserved 10b - 1.1 V 11b - Reserved

24.7.9 SRAM Retention LDO Control (SRAMRETLDO_CNTRL)

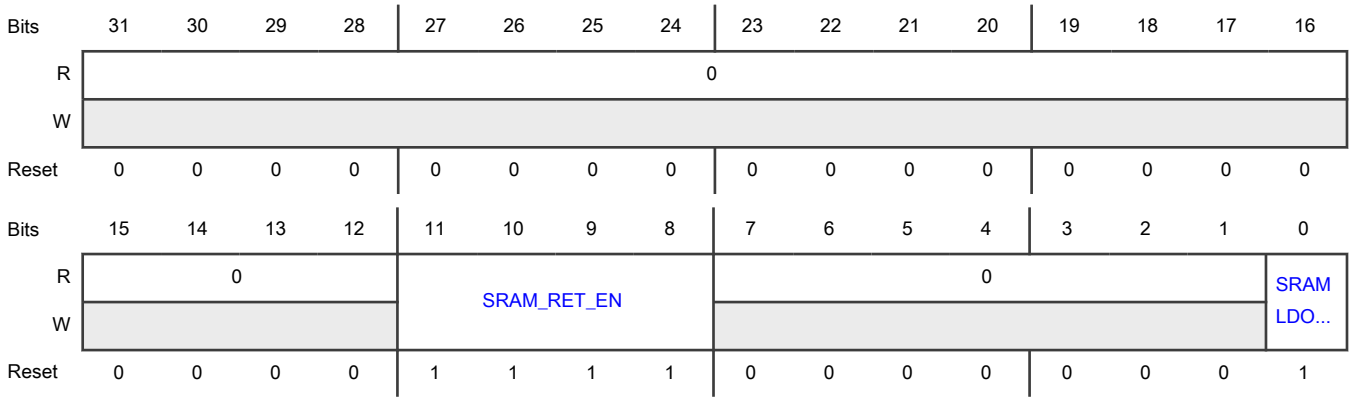
Offset

Register	Offset
SRAMRETLDO_CNTRL	58h

Function

Controls the SRAM retention LDO operation.

Diagram



Fields

Field	Function
31-12 —	Reserved
11-8 SRAM_RET_EN	SRAM Retention Configures the retention of each SRAM array in low-power modes. When SRAMLDO_ON is 1, you must retain at least one SRAM array.

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	When SRAMLDO_ON is 0, SRAM_RET_EN specifies whether the SRAM array remains powered from VDD_CORE. See the chip-specific SPC information for SRAM retention.
7-1 —	Reserved
0 SRAMLDO_ON	SRAM LDO Regulator Enable 0b - Disable 1b - Enable

24.7.10 Active Power Mode Configuration (ACTIVE_CFG)

Offset

Register	Offset
ACTIVE_CFG	100h

Function

Controls the SPC regulators settings in the Active power mode.

Changes to the drive strength or voltage level for any of the LDOs sets the SC[BUSY] flag until SPC changes its state to the new value.

NOTE

The LVDE and HVDE fields reset only with a POR. All other fields reset only with a system reset.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0		0	SYS_HVDE	0	0	SYS_LVDE	CORE_LV...	Reserved	0	BGMODE		0	0	0	
W																
Reset	0	0	0	1	0	0	1	1	0	0	0	1	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0		0	0		0		0	0	0	0	CORELDO_VDD_LVL		0	CORELDO...
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Fields

Field	Function
31-30 —	Reserved
29 —	Reserved
28 SYS_HVDE	System High-Voltage Detection Enable Enables the system high-voltage detection. When SYS_HVDE = 1, you must write a value to BGMODE that enables the bandgap. Only a POR resets this field. 0b - Disable 1b - Enable
27 —	Reserved
26 —	Reserved
25 SYS_LVDE	System Low-Voltage Detection Enable Enables the system low-voltage detection. When SYS_LVDE = 1, you must write a value to BGMODE that enables the bandgap. Only a POR resets this field. 0b - Disable 1b - Enable
24 CORE_LVDE	Core Low-Voltage Detection Enable Enables the core low-voltage detection. When CORE_LVDE = 1, you must write a value to BGMODE that enables the bandgap. Only a POR resets this field. 0b - Disable 1b - Enable
23 —	Reserved
22 —	Reserved
21-20	Bandgap Mode

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
BGMODE	Specifies the bandgap mode configuration. SPC forces BGMODE to 01b to enable the bandgap, if: • A write to disable the bandgap is detected while any regulator is in normal drive strength. • Any of the LVD or HVDs are enabled. 00b - Bandgap disabled 01b - Bandgap enabled, buffer disabled 10b - Bandgap enabled, buffer enabled 11b - Reserved
19 —	Reserved
18 —	Reserved
17-16 —	Reserved
15 —	Reserved
14-13 —	Reserved
12 —	Reserved
11-10 —	Reserved
9-8 —	Reserved
7 —	Reserved
6 —	Reserved
5 —	Reserved

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
4 —	Reserved
3-2 CORELDO_VDD_LVL	LDO_CORE VDD Regulator Voltage Level Selects the LDO_CORE VDD regulator level. If the CORELDO_VDD_DS fields are set to the same value in both the ACTIVE_CFG and LP_CFG registers, the CORELDO_VDD_LVL's in the ACTIVE_CFG and LP_CFG register must be set to the same voltage level settings. You can change the core VDD levels for the LDO_CORE low power regulator only when CORELDO_VDD_DS=1. When switching CORELDO_VDD_DS from low to normal drive strength, ensure CORELDO_VDD_LVL setting is set to the same level that was set prior to switching the CORELDO drive strength (CORELDO_VDD_DS). Otherwise, if the LVDs are enabled, and an unexpected LVD can occur. For example: Initially, CORELDO_VDD_DS is set to 1b (normal drive strength) and CORELDO_VDD_LVL is set to 11b (1.2 V). If the user switches CORELDO_VDD_DS to 0b (low drive strength) and CORELDO_VDD_LVL to 10b (1.1 V), then switching CORELDO drive strength back to normal drive strength by setting CORELDO_VDD_DS to 1b requires that CORELDO_VDD_LVL also be set back to 11b at the same time. This ensures that the supply voltage returns to the level that was used before changing CORELDO to low drive strength. 10b - Regulate to standard drive Mode voltage All other values are reserved.
1 —	Reserved
0 CORELDO_VDD_DS	LDO_CORE VDD Drive Strength Selects the LDO_CORE VDD regulator drive strength **NOTE** When setting CORELDO_VDD_DS to Normal, BGMODE must be programmed to a value that enables the Bandgap. Writes to set drive strength to Low will be ignored if LVD/HVDs are kept enabled. 0b - Low 1b - Normal

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

24.7.11 Active Power Mode Configuration 1 (ACTIVE_CFG1)

Offset

Register	Offset
ACTIVE_CFG1	104h

Function

Enables different analog modules in Active mode. To see what analog modules each of these bits are controlling, see the chip-specific SPC information.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SOC_CNTRL															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SOC_CNTRL															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-0 SOC_CNTRL	Active Config Chip Control Enables analog modules in Active mode. Bit combinations with 0: The associated analog modules are disabled. Bit combinations with 1: The associated analog modules are enabled.

24.7.12 Low-Power Mode Configuration (LP_CFG)

Offset

Register	Offset
LP_CFG	108h

Function

Controls the SPC regulator settings in low-power stop modes.

This register resets only after a POR or LVD event.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0		0	SYS_HVDE	0	0	SYS_LVDE	CORE_LV...	LP_IREF...	0	BGMODE		SRAM LDO...	0	0	0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0		1	0		0		0	0		0	CORELDO_VDD_LVL		0	CORE LDO...
W																
Reset	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0

Fields

Field	Function
31-30 —	Reserved
29 —	Reserved
28 SYS_HVDE	System High Voltage Detect Enable This field resets only after a POR, LVD, or HVD event. When SYS_HVDE = 1, you must use LP_CFG[BGMODE] to enable the bandgap. If you enable the bandgap to support voltage detection, the low-power mode Idd increases. 0b - Disable 1b - Enable
27 —	Reserved
26 —	Reserved
25 SYS_LVDE	System Low Voltage Detect Enable This field resets only after a POR, LVD, or HVD event. When SYS_LVDE = 1, you must use LP_CFG[BGMODE] to enable the bandgap. If you enable the bandgap to support voltage detection, the low-power mode Idd increases. 0b - Disable 1b - Enable
24 CORE_LVDE	Core Low Voltage Detect Enable This field resets only after a POR, LVD, or HVD event.

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	When CORE_LVDE = 1, you must use LP_CFG[BGMODE] to enable the bandgap. If you enable the bandgap to support voltage detection, the low-power mode Idd increases. 0b - Disable 1b - Enable
23 LP_IREFEN	Low-Power IREF Enable You can disable the low-power IREF only when all power domains are in DPDOWN mode. In all other low-power modes, writing to this field has no effect and the low-power IREF is always enabled. See the chip-specific SPC information for the modules that require the low-power IREF to be enabled in DPDOWN mode when you use the module in DPDOWN mode. 0b - Disable for power saving in Deep Power Down mode 1b - Enable
22 —	Reserved
21-20 BGMODE	Bandgap Mode Specifies the bandgap mode configuration. BGMODE will be set to 01b to keep the Bandgap enabled if a write to disable Bandgap is detected while keeping any of the regulators in normal drive strength or if any of the LVD/HVDs are kept enabled. 00b - Bandgap disabled 01b - Bandgap enabled, buffer disabled 10b - Bandgap enabled, buffer enabled 11b - Reserved
19 SRAMLDO_DPD_ON	SRAM_LDO Deep Power Low Power IREF Enable When set keeps the retention SRAM **NOTE** See the chip specific section for the modules that require the low power IREF to be enabled in DPDOWN mode, if the module is being used in DPDOWN mode. 0b - Low Power IREF is disabled for power saving in Deep Power Down mode 1b - Low Power IREF is enabled
18 —	Reserved
17 —	Reserved
16	Reserved

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
—	
15 —	Reserved
14-13 —	Reserved
12 —	Reserved
11-10 —	Reserved
9-8 —	Reserved
7 —	Reserved
6-5 —	Reserved
4 —	Reserved
3-2 CORELDO_VDD_LVL	LDO_CORE VDD Regulator Voltage Level Specifies the LDO_CORE VDD regulator level for low-power sleep modes. If the CORELDO_VDD_DS fields are set to the same value in both the ACTIVE_CFG and LP_CFG registers, the CORELDO_VDD_LVL's in the ACTIVE_CFG and LP_CFG register must be set to the same voltage level settings. You can change the core VDD levels for the LDO_CORE low power regulator only when ACTIVE_CFG[CORELDO_VDD_DS] = 1. So, before entering any of the low-power states (DSLEEP, PDOWN, DPDOWN) with LDO_CORE low power regulator selected (LP_CFG[CORELDO_VDD_DS] = 0), you must use CORELDO_VDD_LVL to select the correct regulation level during ACTIVE run mode. Updating CORELDO_VDD_LVL sets the SC[BUSY] flag. That flag remains set for at least the total time delay that Active Voltage Trim Delay (ACTIVE_VDELAY) specifies. Before changing CORELDO_VDD_LVL, you must wait until the SC[BUSY] flag clears before entering the selected low-power sleep mode. This ensures the correct core VDD voltage levels are set after the chip is in the chosen mode. 00b - Retention voltage

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	10b - Regulate to standard drive Mode voltage 11b - Reserved All other values are reserved.
1 —	Reserved
0 CORELDO_VDD_DS	LDO_CORE VDD Drive Strength Specifies the drive strength of the LDO_CORE VDD regulator. The LDO_CORE regulator is forced to disabled in DPDOWN Sleep mode. If you specify normal drive strength, you must write a value to LP[BGMODE] that enables the bandgap. If you have enabled LVDs or HVDs, and attempt to specify low drive strength, SPC ignores the attempt. 0b - Low 1b - Normal

24.7.13 Low Power Mode Configuration 1 (LP_CFG1)

Offset

Register	Offset
LP_CFG1	10Ch

Function

Enables different analog modules in Low-Power mode.

See the chip-specific SPC information for a list of analog modules that this register controls.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SOC_CNTRL															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SOC_CNTRL															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Fields

Field	Function
31-0 SOC_CNTRL	Low-Power Configuration Chip Control Provides control bits to the chip to enable analog modules in Low Power mode. Bit combinations with 0: The associated analog modules are disabled. Bit combinations with 1: The associated analog modules are enabled.

24.7.14 Low Power Wake-Up Delay (LPWKUP_DELAY)

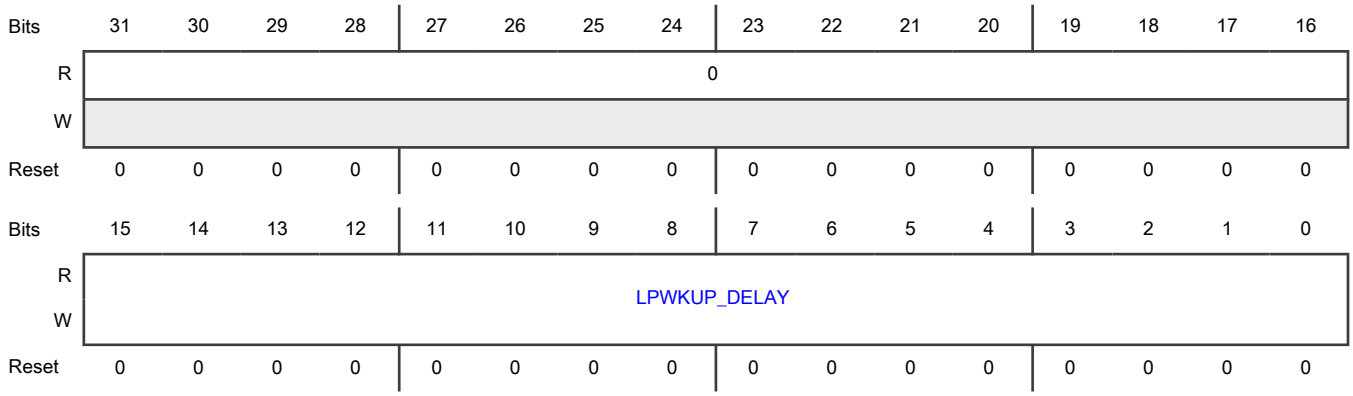
Offset

Register	Offset
LPWKUP_DELAY	120h

Function

Works with the LPREQ pin to extend the wake-up time if the external PMIC or switch needs additional time after wake-up.

Diagram



Fields

Field	Function
31-16 —	Reserved
15-0 LPWKUP_DELAY	Low-Power Wake-Up Delay Specifies the number of SPC timer clock cycles that SPC waits after exiting low-power modes. This field resets only after a POR, LVD, or HVD event. When voltage levels are not the same between ACTIVE mode and Low Power mode, you must write a nonzero value to this field.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

24.7.15 Active Voltage Trim Delay (ACTIVE_VDELAY)

Offset

Register	Offset
ACTIVE_VDELAY	124h

Function

Specifies the delay due to voltage level changes to the regulators in Active mode.

This register resets only after a POR event.

SPC loads the ACTIVE_VDELAY values are loaded from IFR or FUSE after any reset. NXP trims this reset value for the recommended wait time.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	ACTIVE_VDELAY															
W																
Reset	0	0	0	0	0	0	0	0	1	1	0	0	1	0	0	0

Fields

Field	Function
31-16 —	Reserved
15-0 ACTIVE_VDELAY	Active Voltage Delay Specifies the number of SPC timer clock cycles that SPC waits when changing the core-regulator voltage level in Active mode.

24.7.16 Voltage Detect Status (VD_STAT)

Offset

Register	Offset
VD_STAT	130h

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Function

Displays the LVD or HVD captured for:

- Core VDD
- System LVD

This register resets only after a POR event.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0									0	0	0	0	0	0	0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0									0	SYSVDD_...	0	0	0	SYSVDD_...	COREVDD...
W											W1C				W1C	W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-23 —	Reserved
22 —	Reserved
21 —	Reserved
20 —	Reserved
19 —	Reserved
18 —	Reserved
17 —	Reserved
16	Reserved

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
—	
15-7 —	Reserved
6 —	Reserved
5 SYSVDD_HVDF	System HVD Flag Indicates a system HVD event. **NOTE** This field behaves differently for register reads and writes. When reading 0b - Event not detected 1b - Event detected When writing 0b - No effect 1b - Clear the flag
4 —	Reserved
3 —	Reserved
2 —	Reserved
1 SYSVDD_LVDF	System Low-Voltage Detect Flag Indicates a system LVD event. **NOTE** This field behaves differently for register reads and writes. When reading 0b - Event not detected 1b - Event detected When writing 0b - No effect 1b - Clear the flag

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
0 COREVDD_LVDF	Core Low-Voltage Detect Flag Indicates a core LVD event. **NOTE** This field behaves differently for register reads and writes. When reading 0b - Event not detected 1b - Event detected When writing 0b - No effect 1b - Clear the flag

24.7.17 Core Voltage Detect Configuration (VD_CORE_CFG)

Offset

Register	Offset
VD_CORE_CFG	134h

Function

Provides these function:

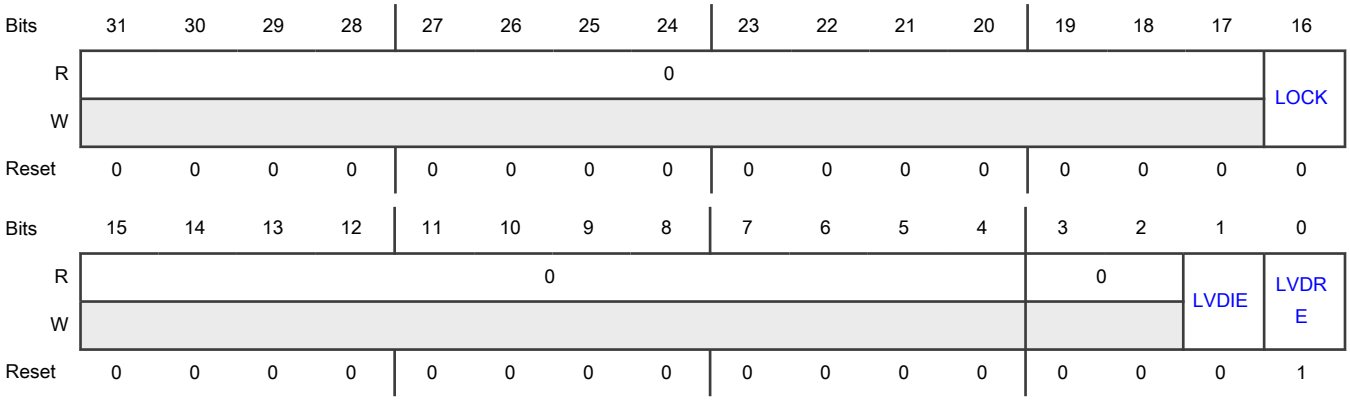
- Enables the core LVD reset and interrupts.

This register resets only after a POR event.

NOTE

If you write 1 to both LVDRE and LVDIE, SPC generates an interrupt after exiting from LVDRE reset. If you don't want this to occur, configure the LVD or HVD so only one is enabled.

Diagram



All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Fields

Field	Function
31-17 —	Reserved
16 LOCK	Core Voltage Detect Reset Enable Lock Allows writing to the LVDRE and HVDRE fields. 0b - Allow 1b - Deny
15-4 —	Reserved
3-2 —	Reserved
1 LVDIE	Core LVD Interrupt Enable Enables the Core LVD (COREVDD_LVDF) event to generate a hardware interrupt. If you write 1 to LVDIE with LVDF set, SPC automatically generates an LVD interrupt. 0b - Disable 1b - Enable
0 LVDRE	Core LVD Reset Enable Enables the core LVD (COREVDD_LVDF) event to generate a hardware reset. Before writing to this field, you must write 0 to LOCK. 0b - Disable 1b - Enable

24.7.18 System Voltage Detect Configuration (VD_SYS_CFG)

Offset

Register	Offset
VD_SYS_CFG	138h

Function

Provides these function:

- Enables the system LVD reset and interrupts.

This register resets only after a POR event.

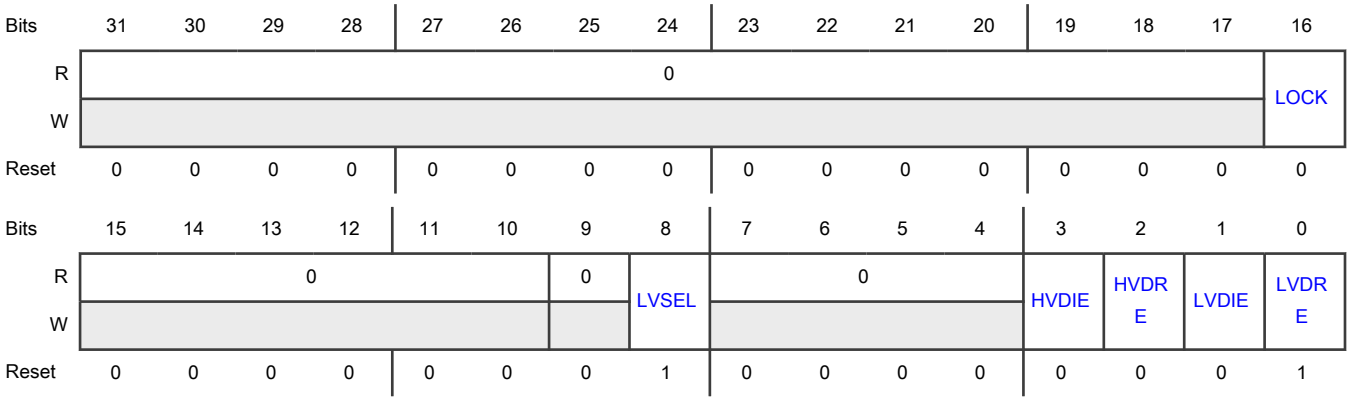
All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

NOTE

If you write 1 to both LVDRE and LVDIE, SPC generates an interrupt after exiting from LVDRE reset. If you don't want this to occur, configure the LVD or HVD so only one is enabled.

Diagram



Fields

Field	Function
31-17 —	Reserved
16 LOCK	System Voltage Detect Reset Enable Lock Allows writing to the LVDRE, HVDRE, and LVSEL fields. 0b - Allow 1b - Deny
15-10 —	Reserved
9 —	Reserved
8 LVSEL	System Low-Voltage Level Select Specifies the LVD trip-point level for the system monitor. See the chip data sheet for the normal- and safe-level values. Before writing to this field, you must write 0 to LOCK. If you want to change LVSEL, you must do this after disabling the LVD reset and interrupt. Otherwise, SPC could generate an LVD due to the LVSEL change. 0b - Normal 1b - Safe
7-4	Reserved

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
—	
3 HVDIE	System HVD Interrupt Enable Enables the system HVD (SYSVDD_HVDF) event to generate a hardware interrupt. If you write 1 to HVDIE with HVDF set, SPC automatically generates an HVD interrupt. 0b - Disable 1b - Enable
2 HVDRE	System HVD Reset Enable Enables the system HVD (SYSVDD_HVDF) event to generate a hardware reset. Before writing to this field, you must write 0 to LOCK. 0b - Disable 1b - Enable
1 LVDIE	System LVD Interrupt Enable Enables the system LVD (SYSVDD_LVDF) event to generate a hardware interrupt. If you write 1 to LVDIE with LVDF set, SPC automatically generates an LVD interrupt. 0b - Disable 1b - Enable
0 LVDRE	System LVD Reset Enable Enables the system LVD (SYSVDD_LVDF) event to generate a hardware reset. Before writing to this field, you must write 0 to LOCK. 0b - Disable 1b - Enable

24.7.19 External Voltage Domain Configuration (EVD_CFG)

Offset

Register	Offset
EVD_CFG	140h

Function

Isolates signals from external voltage domains under certain conditions.

Chip pins supply the external voltage domains. These domains are controlled at the board level. This register allows you to control internal isolations for signals from these domains if either of the following conditions is true:

- They are not powered on the board.
- They are powered off externally in low-power modes (using the SPC_LPREQ pin).

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Any logic in the isolated voltage domain also resets, because the voltage domain is assumed to be powered off. This reset does not impact any VDD_CORE internal power domains or registers—only I/O pads, analog components, or both.

This register resets only after a POR event.

Each bit of the EVDISO, EVDLPISO, and EVDSTAT fields corresponds to one of the I/O or analog supplies according to Table 150.

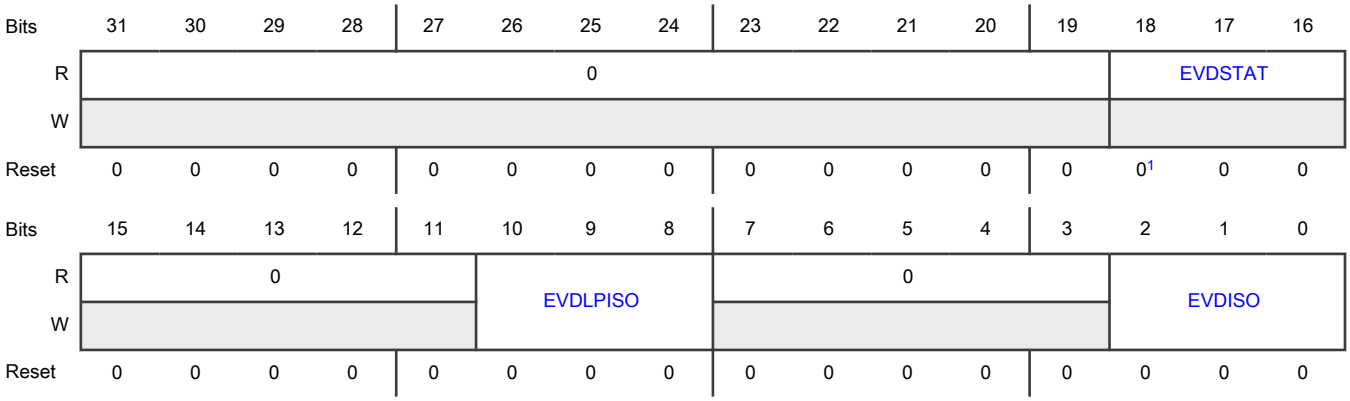
Table 150. Mapping of bits to supplies

Bit position in field	Description
0	VDD
1	VDD_USB
2	Reserved

NOTE

Bit 0 VDD should not be set in EVD_CFG[EVDISO] and it should only be set in EVD_CFG[EVDLPISO] when moving to Deep Power Down mode.

Diagram



1. The reset value of this field can change based on external voltage conditions.

Fields

Field	Function
31-19 —	Reserved
18-16 EVDSTAT	External Voltage Domain Status Indicates the status of the external voltage domains. See Table 150 for the mapping of each bit of this field. Each bit's value has the following meaning: 0b - Isolated or not powered 1b - Not isolated
15-11 —	Reserved

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
10-8 EVDLPISO	External Voltage Domain Low-Power Isolation Isolates the external voltage domain in low-power modes. Use this field if you use the SPC_LPREQ pin to power off any voltage domain in low-power modes. See Table 150 for the mapping of each bit of this field. Each bit's value has the following meaning: 0b - Not isolated 1b - Isolated
7-3 —	Reserved
2-0 EVDISO	External Voltage Domain Isolation Isolates the external voltage domain. Use this field if any voltage domain is always powered off on the board, or whenever a voltage domain is powered off under software control. See Table 150 for the mapping of each bit of this field. Each bit's value has the following meaning: 0b - Not isolated 1b - Isolated

24.7.20 LDO_CORE Configuration (CORELDO_CFG)

Offset

Register	Offset
CORELDO_CFG	300h

Function

Configures LDO_CORE.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0				0				0	0		CORELDO_SPARE				
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0[1]	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0							0	0				0	0	0	0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

1. The bits reset solely on a POR/LVD/HVD event.

Fields

Field	Function
31-29 —	Reserved
28-24 —	Reserved
23 —	Reserved
22-21 —	Reserved
20-16 CORELDO_SPARE	CORELDO_SPARE DAC/ATS Bandgap Buffer Enable register. 0_0000b - TBD. 0_0001b - TBD.
15-9 —	Reserved
8 —	Reserved
7-4 —	Reserved
3 —	Reserved
2 —	Reserved
1 —	Reserved
0 —	Reserved

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Chapter 25
Standard Counter/Timers (CTIMER)

25.1 Chip-specific CTIMER information

Table 151. Reference links to related information

Topic	Related module	Reference
Full description	CTIMER	CTIMER
System memory map		Memory map
Clocking		Clock distribution
Power Management		Power management
Signal multiplexing	Port control	Signal multiplexing
Peripheral Input Mux	INPUTMUX	Peripheral input multiplexing

25.1.1 Module instances

This chip has two instances of the CTIMER module, CTIMER0 and CTIMER1. Each timer has up to 4x capture and 4x match registers with corresponding inputs and outputs.

CTIMER supports async function clock. The async function clock speed can be higher than APB bus clock.

25.1.2 Signals

The tables below give a summary of each of the Timer/Counter related pins and the recommended PORT settings. Also note that different part number and package variations may provide different CTIMER related pin functions.

Table 152. Timer/Counter pin description

Pin	Type	Description
CTIMER0_CAP3:0 CTIMER1_CAP3:0	Input	Capture Signals- A transition on a capture pin can be configured to load one of the Capture Registers with the value in the Timer Counter and optionally generate an interrupt. Capture functionality can be selected from a number of pins. Timer/Counter block can select a capture signal as a clock source instead of the APB bus clock. For more details, see CTCR register.
CT0_MAT3:0 CT1_MAT3:0	Output	External Match Output - When a match register (MR3:0) equals the timer counter (TC) this output can either toggle, go low, go high, or do nothing. The External Match Register (EMR) controls the functionality of this output. Match Output functionality can be selected on a number of pins in parallel.

25.1.3 External global enable

When the timer is enabled by the CTIMER global start register in SYSCON, (register CTIMERGLOBALSTARTEN in System Controller (SYSCON)), it is equivalent to writing 1 to the TCR[CEN] bit in CTIMER. The external global start register is used to enable multiple timers at the same time. This operation is allowed by writing a 1 to each Timer's TCR[AGCEN]

All information provided in this document is subject to legal disclaimers.
© 2026 NXP B.V. All rights reserved.

bit. If the Timer control register's CEN bit is already 1, the external global start enable register has no effect (See register CTIMERGLOBALSTARTEN in System Controller (SYSCON)).

25.1.4 Peripheral input multiplexers

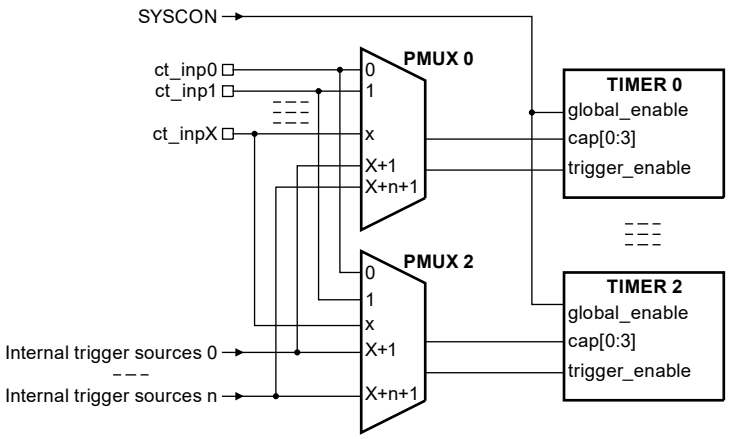


Figure 82. Peripheral input multiplexers for CTIMER instances

25.1.5 Initialization

1. Select a clock source for the CTIMER using CTIMER0 clock selection control (MRCC_CTIMER0_CLKSEL) and CTIMER1 clock selection control (MRCC_CTIMER1_CLKSEL) registers.
2. Enable the clock to the CTIMER via the CTIMER0_CLK_EN, CTIMER1_CLK_EN fields. This enables the register interface and the peripheral function clock.
3. Clear the CTIMER peripheral reset using the (Peripheral Reset Control 0 (MRCC_GLB_RST0) register.
4. Each CTIMER provides interrupts to the NVIC. See MCR and CCR registers in the CTIMER register section for match and capture events. For interrupt connections, see Interrupt channel assignments.
5. Select timer pins and pin modes as needed through the relevant PORT registers.

25.2 Overview

Each CTIMER is designed to count cycles of the CTIMER function clock or an externally supplied clock. A CTIMER can optionally generate an interrupt or perform other actions at specified timer values based on the settings of Match (MR0 - MR3). Each CTIMER also includes capture input pins (see Capture mode) to capture the timer value when an input signal transitions, optionally generating an interrupt.

In PWM mode, three match registers can be used to provide a single-edge controlled PWM output on the match output pins. One match register is used to control the PWM cycle length. All match registers can optionally be auto-reloaded from a companion shadow register (see Match Shadow (MSR0 - MSR3)) whenever the counter is reset to zero. This feature permits modifying the match values for the next counter cycle without disrupting PWM waveforms during the current cycle. When enabled, match reload occurs whenever the counter is reset, either because of a match event or by writing 1 to TCR[CRST].

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

25.2.1 Block diagram

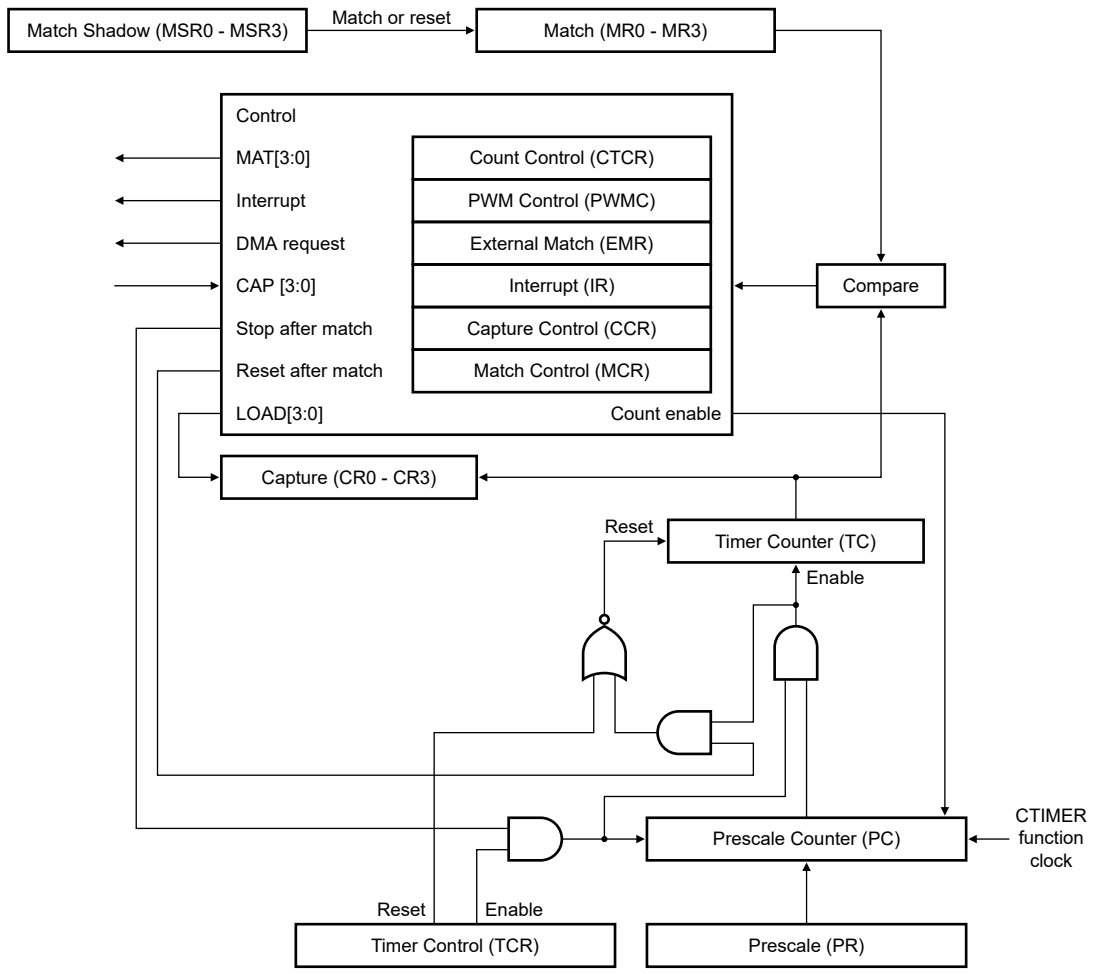


Figure 83. Block diagram

25.2.2 Features

- Contains a 32-bit CTIMER with a programmable 32-bit prescaler (the timers include external capture and match pin connections)
- Contains Timer Control (TCR) and External trigger enable, either of which you can use to enable each timer (you can also use an external global start enable register to globally enable one or more timers)
- Performs counter or timer operations
- Selects the function clocks that may be asynchronous to other system clocks
- Includes up to four 32-bit captures (see Capture mode) to take a snapshot of the timer value when an input signal transitions (a capture event may also optionally generate an interrupt, and the number of capture inputs available for each timer on a chip may vary)
- Enables you to configure the timer and prescaler to be cleared on a designated capture event (this feature permits easy pulse-width measurement by clearing the timer on the leading edge of an input pulse, capturing the timer value on the trailing edge)
- Includes Match (MR0 - MR3) to allow:
  - Continuous operation with optional interrupt generation after a match occurs

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

  - Optional auto-reload from Match Shadow (MSR0 - MSR3) when the counter is reset
  - Stop timer after a match with optional interrupt generation
  - Reset timer after a match with optional interrupt generation
- Generates up to four external outputs corresponding to Match (MR0 - MR3), for each timer, with the following capabilities (the number of match outputs available for each timer on a chip may vary):
  - Go low when matched
  - Go high when matched
  - Toggle when matched
  - Do nothing
- Enables you to configure up to four match registers for PWM operation, allowing up to three single-edged controlled PWM outputs (the number of match outputs available for each timer on a chip may vary)
- Uses up to two match registers to generate DMA requests

25.3 Functional description

Figure 84 shows a timer configured to reset the count and generate an interrupt after a match. The value of Prescale (PR) is 2 and the value of Match (MR0 - MR3) is 6. At the end of the timer cycle where the match occurs, the timer count is reset and gives a full-length cycle to the match value. The interrupt indicating that a match has occurred is generated in the next clock after the timer reaches the match value. Timer mode is selected in this case.

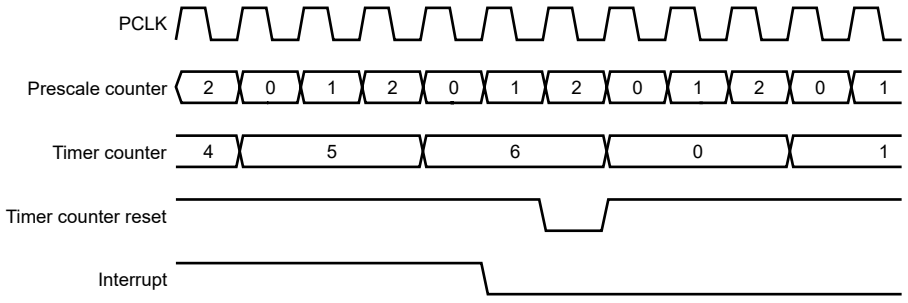


Figure 84. Timer cycle with PR = 2, MR*x* = 6, and both interrupt and reset enabled after match

Figure 85 shows a timer configured to stop and generate an interrupt after a match. The value of Prescale (PR) is 2 and the value of Match (MR0 - MR3) is 6. In the next clock after the timer reaches the match value, TCR[CEN] becomes 0 and the interrupt indicating that a match has occurred is generated.

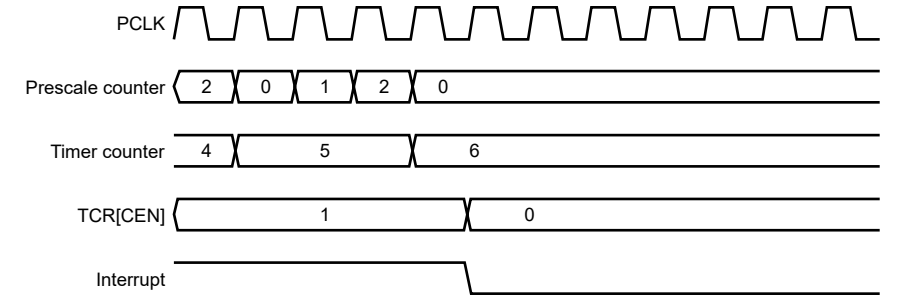


Figure 85. Timer cycle with PR = 2, MR*x* = 6, and both interrupt and stop enabled after match

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

25.3.1 Operation sections

All single-edge controlled PWM outputs go low at the beginning of each PWM cycle (timer is set to zero) unless their match value is zero.

Each PWM output goes high when its match value is reached. If no match occurs (that is, the match value is greater than the PWM cycle length), the PWM output remains continuously low.

If a match value larger than the PWM cycle length is written to Match (MR0 - MR3), and the PWM signal is already high, the PWM signal is cleared at the start of the next PWM cycle.

If a match register contains the same value as the timer reset value (the PWM cycle length), the PWM output is reset to low on the next clock tick after the timer reaches the match value. Therefore, the PWM output always consists of one clock tick wide positive pulse with a period that the PWM cycle length determines (that is, the timer reload value).

If Match (MR0 - MR3) = 0, the PWM output goes to high the first time the timer goes back to zero, and remains high continuously.

The following figure shows sample PWM waveforms with a PWM cycle length of 100 that MR3 specifies and PWM outputs that PWM Control (PWMC) enables. Counter mode is selected in this case.

NOTE

When you select match outputs to perform as PWM outputs, you must write 0 to MCR[MR*n*R] and MCR[MR*n*S], except to the match register setting the PWM cycle length. For this register, write 1 to MCR[MR*n*R] to enable the timer reset when the timer value matches the value of the corresponding match register.

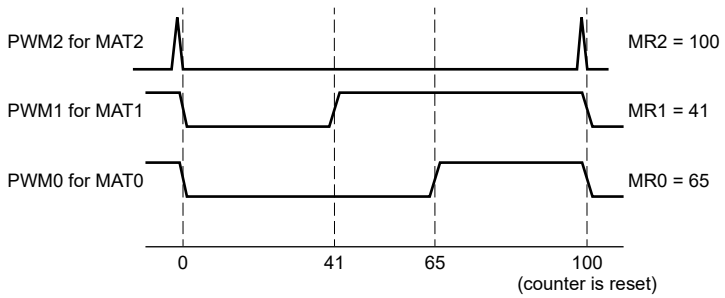


Figure 86. Sample PWM waveforms with a PWM cycle length of 100 (selected by MR3) and MAT3:0 enabled as PWM outputs by the PWM Control register.

25.3.2 Mode sections

25.3.2.1 Capture mode

You can configure a CAP to load Capture (CR0 - CR3) with the value in the counter or timer and optionally generate an interrupt. One of the pins generates the capture signal with a capture function. Each CAP is connected to one capture channel of the timer.

CTIMER can select a CAP as a clock source instead of selecting the APB CTIMER function clock. See Count Control (CTCR) for more information.

25.3.2.2 Match mode

When the value of a match register (Match (MR0 - MR3)) equals that of Timer Counter (TC), the corresponding match output can either toggle, go low, go high, or do nothing. External Match (EMR) and PWM Control (PWMC) control the functionality of this output.

25.3.3 External trigger enable

When trigger_enable is asserted (rising edge), it is equivalent to writing 1 to TCR[CEN]. If TCR[CEN] is already 1, trigger_enable rising edge events have no effect.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

25.3.4 Clocking

Table 153. CTIMER clocks

Type of clock	Description
Functional Clock(ctclk)	Asynchronous to the bus clock, provide clock to perform counter or timer operations.
Bus Clock(pclk)	The bus clock is only used for bus accesses to the control and configuration registers.

25.3.5 Reset

TCR[CRST] resets Timer Counter (TC) and Prescale Counter (PC).

25.3.6 Interrupts

This module has capture and match interrupts. See the Interrupt (IR)

25.3.7 DMA

DMA requests are generated when the value of Timer Counter (TC) either matches the value of MR0 or MR1 (see Match (MR0 - MR3) for more information). This DMA request is not connected to the operation of the match outputs (see Match mode) controlled by External Match (EMR). Each match sets a DMA request flag, which is connected to the DMA controller. You must configure the DMA controller correctly.

When a timer is initially set up to generate a DMA request, the request may already be asserted before a match condition occurs. An initial DMA request may be avoided by writing 1 to the interrupt flag location twice, as if clearing a timer interrupt. See Interrupt (IR) for more information. A DMA request is cleared automatically when the DMA controller manages it.

NOTE

Timer DMA requests are generated whenever the timer value is equal to the related match register value. DMA requests are always generated when the timer is running, unless the match register value is higher than the upper count limit of the timer. It is important not to select and enable timer DMA requests in the DMA block unless you configure the timer correctly to generate valid DMA requests.

25.4 External signals

Table 154. CTIMER pin descriptions

Pin	Type	Description
CTIMER*m*_CAP*n*	Input	Capture signals—You can configure a transition on a capture pin to load one of the capture registers (Capture (CR0 - CR3)) with the value of Timer Counter (TC) and optionally generate an interrupt. You can select the capture functionality for a number of pins. CTIMER can select a capture signal as a clock source instead of the APB bus clock. See Count Control (CTCR) for more information.
CTIMER*m*_MAT*n*	Output	External match output—When the value of a match register (Match (MR0 - MR3)) equals the value of Timer Counter (TC), this output can either toggle, go low, go high, or do nothing. External Match (EMR) controls the functionality of this output. You can select the match output functionality for a number of pins in parallel.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

25.4.1 Multiple capture (CAP) and match (MAT) pins

You can configure the pins of the chip to be used for this module. These pins are generally reused, and you can connect several internal module pins selectively. However, you can connect only one internal pin at a time.

NOTE

Match conditions may be used internally without the use of a chip's pin.

25.5 Initialization

See the Chip-specific CTIMER information for the initialization section for more details or steps.

25.6 Application information

CTIMER register descriptions discusses the following:

- Interval timer for counting internal events
- PWM outputs
- Pulse width demodulator via capture inputs
- Free running timer

25.7 CTIMER register descriptions

25.7.1 CTIMER memory map

CTIMER0 base address: 4000_4000h

CTIMER1 base address: 4000_5000h

Offset	Register	Width (In bits)	Access	Reset value
0h	Interrupt (IR)	32	RW	See section
4h	Timer Control (TCR)	32	RW	See section
8h	Timer Counter (TC)	32	RW	0000_0000h
Ch	Prescale (PR)	32	RW	0000_0000h
10h	Prescale Counter (PC)	32	RW	0000_0000h
14h	Match Control (MCR)	32	RW	See section
18h - 24h	Match (MR0 - MR3)	32	RW	0000_0000h
28h	Capture Control (CCR)	32	RW	See section
2Ch - 38h	Capture (CR0 - CR3)	32	R	0000_0000h
3Ch	External Match (EMR)	32	RW	See section
70h	Count Control (CTCR)	32	RW	See section
74h	PWM Control (PWMC)	32	RW	See section
78h - 84h	Match Shadow (MSR0 - MSR3)	32	RW	0000_0000h

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

25.7.2 Interrupt (IR)

Offset

Register	Offset
IR	0h

Function

Provides flags for match interrupts and capture interrupts. If an interrupt is generated, it sets the corresponding flag in this register. Otherwise, the flag remains cleared. Writing 1 to the corresponding IR field resets the interrupt. Writing 0 has no effect. Clearing an interrupt for a timer match also clears any corresponding DMA request. You can write to this register to clear interrupts, and can read this register to identify which interrupt sources are pending.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved								CR3INT	CR2INT	CR1INT	CR0INT	MR3INT	MR2INT	MR1INT	MR0INT
W									W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C
Reset	u	u	u	u	u	u	u	u	0	0	0	0	0	0	0	0

Fields

Field	Function
31-8 —	Reserved
7-4 CRnINT	Flag for Capture Channel n Event Controls an interrupt flag for capture channel *n* events.
3-0 MRnINT	Interrupt Flag for Match Channel n Event Controls an interrupt flag for match channel *n* events.

25.7.3 Timer Control (TCR)

Offset

Register	Offset
TCR	4h

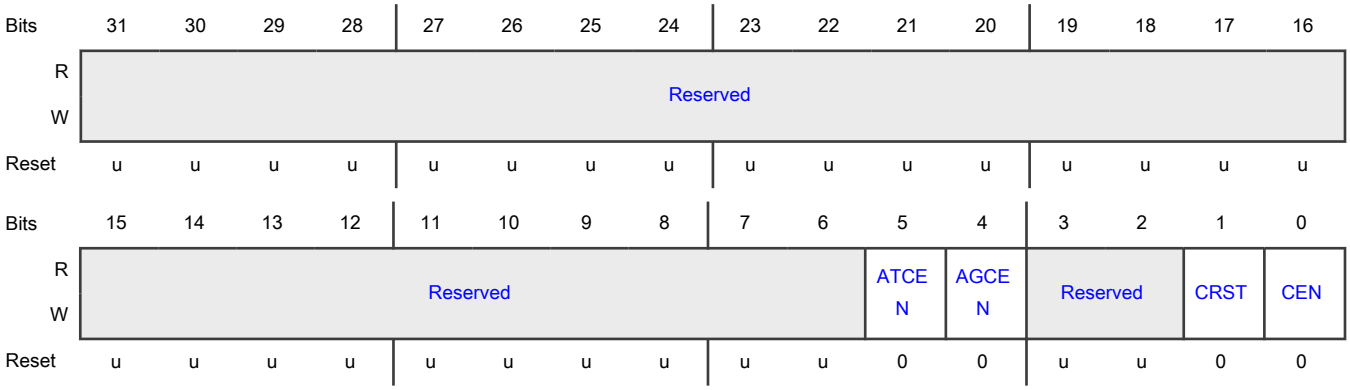
All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Function

Controls timer counter functions. You can disable or reset Timer Counter (TC) by using this register.

Diagram



Fields

Field	Function
31-6 —	Reserved
5 ATCEN	Allow Trigger Count Enable Enables the input trigger. If this field is 1, it allows the input trigger_enable = 1 action to take effect (see External trigger enable). If this field is 0, the action is not allowed. 0b - Disable 1b - Enable
4 AGCEN	Allow Global Count Enable Enables the global count. If this field is 1, it allows the input global_enable = 1 action to take effect. If this field = 0, the action is not allowed. 0b - Disable 1b - Enable
3-2 —	Reserved
1 CRST	Counter Reset Enable Enables the counter reset. If this field is 1, Timer Counter (TC) and Prescale Counter (PC) are synchronously reset after the next positive edge of the CTIMER function clock. The counters remain reset until this field becomes 0. If this field is 0, no effect takes place. 0b - Disable 1b - Enable
0	Counter Enable

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
CEN	Enables the counters. If this field is 1, Timer Counter (TC) and Prescale Counter (PC) are enabled. When an external trigger enables the timer or the external global start enable register enables it, this field becomes 1 automatically. 0b - Disable 1b - Enable

25.7.4 Timer Counter (TC)

Offset

Register	Offset
TC	8h

Function

Specifies the timer counter.

This register is incremented when Prescale Counter (PC) reaches its terminal count. Unless it is reset before reaching its upper limit, Timer Counter (TC) counts up through the value FFFF_FFFFh, and then wraps back to the value 0000_0000h. This event does not cause an interrupt, but you can use Match (MR0 - MR3) to detect an overflow if needed.

Timer Counter (TC) is incremented after every Prescale (PR) + 1 cycle of the CTIMER function clock. Timer Control (TCR) controls the functioning of this register.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	TCVAL															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	TCVAL															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-0 TCVAL	Timer Counter Value Specifies the value of the timer counter.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

25.7.5 Prescale (PR)

Offset

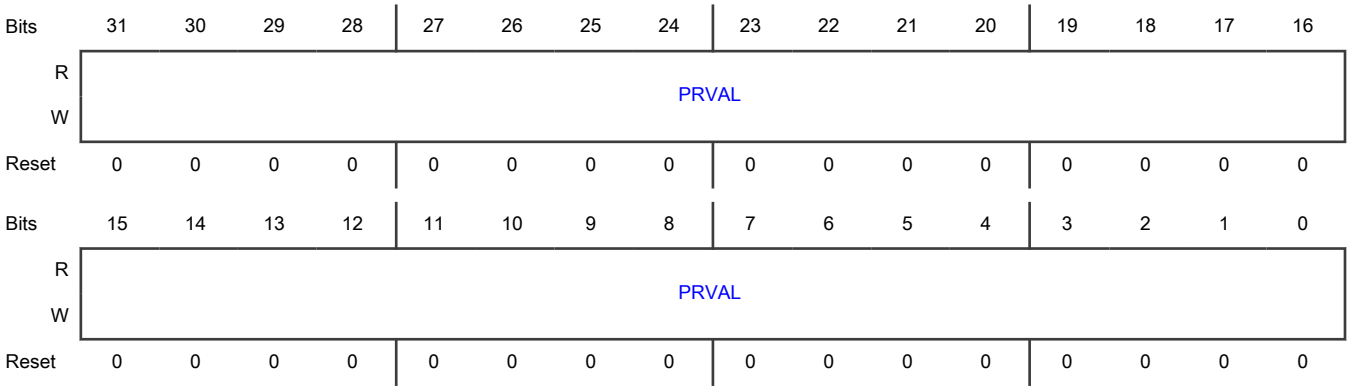
Register	Offset
PR	Ch

Function

Specifies the prescale value.

If Prescale Counter (PC) is equal to the prescale value, the next clock timer increments Timer Counter (TC) and turns the value of Prescale Counter (PC) to 0.

Diagram



Fields

Field	Function
31-0 PRVAL	Prescale Reload Value Specifies the value of the prescale reload.

25.7.6 Prescale Counter (PC)

Offset

Register	Offset
PC	10h

Function

Controls the division of the CTIMER function clock by some constant value before it is applied to Timer Counter (TC). This allows control of the relationship between the resolution of the timer and the maximum time before the timer overflows.

Prescale Counter (PC) is incremented on every CTIMER function clock. When it reaches the value stored in Prescale (PR), Timer Counter (TC) is incremented and Prescale Counter (PC) is reset on the next CTIMER function clock. This causes Timer Counter (TC) to increment on every CTIMER function clock when Prescale (PR) = 0, every two CTIMER function clocks when Prescale (PR) = 1, and so on.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

You can use the bus interface to observe and control Prescale Counter (PC).

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	PCVAL															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PCVAL															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-0 PCVAL	Prescale Counter Value Specifies the value of the prescale counter.

25.7.7 Match Control (MCR)

Offset

Register	Offset
MCR	14h

Function

Controls the operations that are performed when one of the match registers (Match (MR0 - MR3)) matches Timer Counter (TC).

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R					MR3RL	MR2RL	MR1RL	MR0RL								
W	Reserved								Reserved							
Reset	u	u	u	u	0	0	0	0	u	u	u	u	u	u	u	u

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R					MR3S	MR3R	MR3I	MR2S	MR2R	MR2I	MR1S	MR1R	MR1I	MR0S	MR0R	MR0I
W	Reserved															
Reset	u	u	u	u	0	0	0	0	0	0	0	0	0	0	0	0

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Fields

Field	Function
31-28 —	Reserved
27-24 MRnRL	Reload MR Reloads with the contents of the match n shadow register (see Match Shadow (MSR0 - MSR3)) when Timer Counter (TC) is reset to zero (either via a match event or by writing 1 to TCR[CRST]). 0b - Does not reload 1b - Reloads
23-12 —	Reserved
11 MR3S	Stop on MR3 Stops Timer Counter (TC) and Prescale Counter (PC), and turns TCR[CEN] to 0 if MR3 matches Timer Counter (TC). 0b - Does not stop 1b - Stops
10 MR3R	Reset on MR3 Resets Timer Counter (TC) if MR3 matches its value. 0b - Does not reset 1b - Resets
9 MR3I	Interrupt on MR3 Generates an interrupt when MR3 matches the value in Timer Counter (TC). 0b - Does not generate 1b - Generates
8 MR2S	Stop on MR2 Stops Timer Counter (TC) and Prescale Counter (PC), and turns TCR[CEN] to 0 if MR2 matches Timer Counter (TC). 0b - Does not stop 1b - Stops
7 MR2R	Reset on MR2 Resets Timer Counter (TC) if MR2 matches its value. 0b - Does not reset 1b - Resets
6	Interrupt on MR2

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
MR2I	Generates an interrupt when MR2 matches the value in Timer Counter (TC). 0b - Does not generate 1b - Generates
5 MR1S	Stop on MR1 Stops Timer Counter (TC) and Prescale Counter (PC), and turns TCR[CEN] to 0 if MR1 matches Timer Counter (TC). 0b - Does not stop 1b - Stops
4 MR1R	Reset on MR1 Resets Timer Counter (TC) if MR1 matches its value. 0b - Does not reset 1b - Resets
3 MR1I	Interrupt on MR1 Generates an interrupt when MR1 matches the value in Timer Counter (TC). 0b - Does not generate 1b - Generates
2 MR0S	Stop on MR0 Stops Timer Counter (TC) and Prescale Counter (PC), and turns TCR[CEN] to 0 if MR0 matches Timer Counter (TC). 0b - Does not stop 1b - Stops
1 MR0R	Reset on MR0 Resets Timer Counter (TC) if MR0 matches its value. 0b - Does not reset 1b - Resets
0 MR0I	Interrupt on MR0 Generates an interrupt when MR0 matches the value in Timer Counter (TC). 0b - Does not generate 1b - Generates

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

25.7.8 Match (MR0 - MR3)

Offset

Register	Offset
MR0	18h
MR1	1Ch
MR2	20h
MR3	24h

Function

Continuously compares the value of Match (MR0 - MR3) with the value of TC[TCVAL]. When both the values are equal, actions such as the following can be triggered automatically:

- Generation of an interrupt
- Resetting of Timer Counter (TC)
- Stopping of the timer

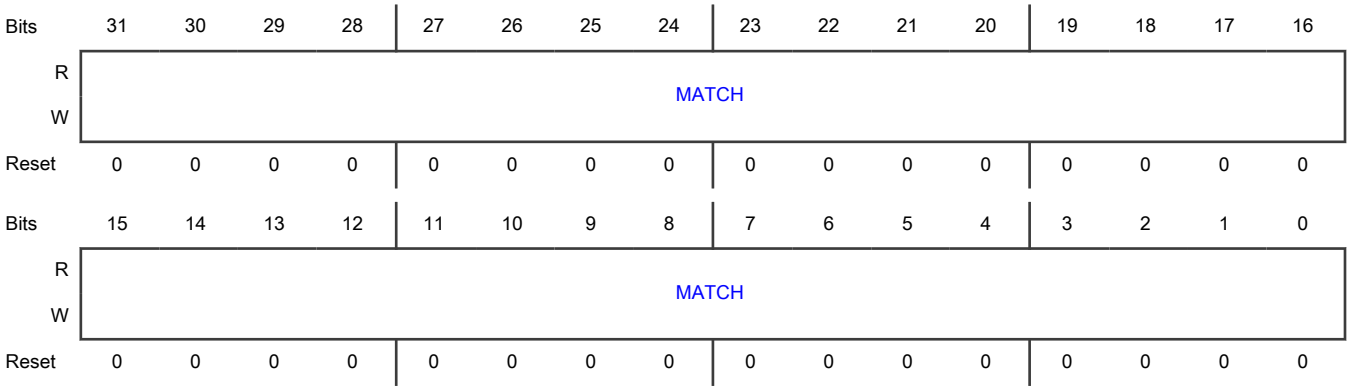
The settings defined in Match Control (MCR) control these actions.

If MCR[MRnRL] = 1, Match (MR0 - MR3) is automatically reloaded with the current contents of its corresponding Match Shadow (MSR0 - MSR3) whenever Timer Counter (TC) becomes 0. This transfer takes place on the same clock edge that advances Timer Counter (TC) to 0.

NOTE

Timer Counter (TC) resets in response to an occurrence of a match on MATCH being used to set the cycle counter rate. A reset can also occur if you write 1 to TCR[CRST].

Diagram



Fields

Field	Function
31-0 MATCH	Timer Counter Match Value Triggers an action automatically when the value of this field is the same as that of TC[TCVAL].

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

25.7.9 Capture Control (CCR)

Offset

Register	Offset
CCR	28h

Function

Controls whether one of the capture registers (Capture (CR0 - CR3)) is loaded with the value in Timer Counter (TC) when the capture event occurs. The register also controls whether the capture event generates an interrupt. When you configure both the rising and falling bits at the same time, which is a valid configuration, it results into a capture event for both edges. In the description below, "*n*" represents the timer number, 0 or 1.

NOTE

If you select Counter mode for a particular CAP*n* input in Count Control (CTCR), you must program the fields in this register as 0. You can select both capture and interrupt for the other CAP*n* inputs.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved				CAP3I	CAP3 FE	CAP3 RE	CAP2I	CAP2 FE	CAP2 RE	CAP1I	CAP1 FE	CAP1 RE	CAP0I	CAP0 FE	CAP0 RE
Reset	u	u	u	u	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-12 —	Reserved
11 CAP3I	Generate Interrupt on Channel 3 Capture Event Generates an interrupt on channel 3 capture event using a CR3 load. 0b - Does not generate 1b - Generates
10 CAP3FE	Falling Edge of Capture Channel 3 Loads CR3 with the contents of Timer Counter (TC) when you write 1 and then 0 to this field.

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	0b - Does not load 1b - Loads
9 CAP3RE	Rising Edge of Capture Channel 3 Loads CR3 with the contents of Timer Counter (TC) when you write 0 and then 1 to this field. 0b - Does not load 1b - Loads
8 CAP2I	Generate Interrupt on Channel 2 Capture Event Generates an interrupt on channel 2 capture event using a CR2 load. 0b - Does not generate 1b - Generates
7 CAP2FE	Falling Edge of Capture Channel 2 Loads CR2 with the contents of Timer Counter (TC) when you write 1 and then 0 to this field. 0b - Does not load 1b - Loads
6 CAP2RE	Rising Edge of Capture Channel 2 Loads CR2 with the contents of Timer Counter (TC) when you write 0 and then 1 to this field. 0b - Does not load 1b - Loads
5 CAP1I	Generate Interrupt on Channel 1 Capture Event Generates an interrupt on channel 1 capture event using a CR1 load. 0b - Does not generates 1b - Generates
4 CAP1FE	Falling Edge of Capture Channel 1 Loads CR1 with the contents of Timer Counter (TC) when you write 1 and then 0 to this field. 0b - Does not load 1b - Loads
3 CAP1RE	Rising Edge of Capture Channel 1 Loads CR1 with the contents of Timer Counter (TC) when you write 0 and then 1 to this field. 0b - Does not load 1b - Loads
2	Generate Interrupt on Channel 0 Capture Event

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
CAP0I	Generates an interrupt on channel 0 capture event using a CR0 load. 0b - Does not generate 1b - Generates
1 CAP0FE	Falling Edge of Capture Channel 0 Loads CR0 with the contents of Timer Counter (TC) when you write 1 and then 0 to this field. 0b - Does not load 1b - Loads
0 CAP0RE	Rising Edge of Capture Channel 0 Loads CR0 with the contents of Timer Counter (TC) when you write 0 and then 1 to this field. 0b - Does not load 1b - Loads

25.7.10 Capture (CR0 - CR3)

Offset

Register	Offset
CR0	2Ch
CR1	30h
CR2	34h
CR3	38h

Function

Works with a capture channel and may load with the value of Timer Counter (TC) when a specified event occurs on the signal defined for that capture channel. The signal could originate from an external pin or an internal source.

This register determines whether the capture function is enabled, and whether a capture event happens on the rising edge, falling edge, or on both edges of the associated signal. The register is loaded with the value of Timer Counter (TC) when there is an event on the CAP*n* input.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	CAP															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CAP															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-0 CAP	Timer Counter Capture Value Determines whether the capture function is enabled and defines the edge on which the capture function occurs.

25.7.11 External Match (EMR)

Offset

Register	Offset
EMR	3Ch

Function

Provides both control and status of the external match pins. If the match outputs (see Match mode) are configured as PWM outputs, PWM rules determine the function of this register.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved				EMC3		EMC2		EMC1		EMC0		EM3	EM2	EM1	EM0
W																
Reset	u	u	u	u	0	0	0	0	0	0	0	0	0	0	0	0

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Fields

Field	Function
31-12 —	Reserved
11-10 EMC3	External Match Control 3 Determines the functionality of EM3: • If this field is 1, the corresponding external match field or output becomes 0 (if pinned out MAT3 pin is low). • If this field is 10, the corresponding external match field or output becomes 1 (if pinned out MAT3 pin is high). • If this field is 11, it toggles the corresponding external match field or output. 00b - Does nothing 01b - Goes low 10b - Goes high 11b - Toggles
9-8 EMC2	External Match Control 2 Determines the functionality of EM2: • If this field is 1, the corresponding external match field or output becomes 0 (if pinned out MAT2 pin is low). • If this field is 10, the corresponding external match field or output becomes 1 (if pinned out MAT2 pin is high). • If this field is 11, it toggles the corresponding external match field or output. 00b - Does nothing 01b - Goes low 10b - Goes high 11b - Toggles
7-6 EMC1	External Match Control 1 Determines the functionality of EM1: • If this field is 1, the corresponding external match field or output becomes 0 (if pinned out MAT1 pin is low). • If this field is 10, the corresponding external match field or output becomes 1 (if pinned out MAT1 pin is high). • If this field is 11, it toggles the corresponding external match field or output. 00b - Does nothing 01b - Goes low 10b - Goes high

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	11b - Toggles
5-4 EMC0	External Match Control 0 Determines the functionality of EM0: • If this field is 1, the corresponding external match field or output becomes 0 (if pinned out MAT0 pin is low). • If this field is 10, the corresponding external match field or output becomes 1 (if pinned out MAT0 pin is high). • If this field is 11, it toggles the corresponding external match field or output. 00b - Does nothing 01b - Goes low 10b - Goes high 11b - Toggles
3 EM3	External Match 3 Reflects the state of output MAT3—whether this output is connected to a pin. If a match occurs between Timer Counter (TC) and MR3 (see Match (MR0 - MR3)), this field can either toggle, go low, go high, or do nothing, as selected by EMC3. This field is driven to the MAT pins if the match function is selected via IOPCTL.Writing to this field can directly change the MAT output state. 0b - Low 1b - High
2 EM2	External Match 2 Reflects the state of output MAT2—whether this output is connected to a pin. If a match occurs between Timer Counter (TC) and MR2 (see Match (MR0 - MR3)), this field can either toggle, go low, go high, or do nothing, as selected by EMC2. This field is driven to the MAT pins if the match function is selected via IOPCTL.Writing to this field can directly change the MAT output state. 0b - Low 1b - High
1 EM1	External Match 1 Reflects the state of output MAT1—whether this output is connected to a pin. If a match occurs between Timer Counter (TC) and MR1 (see Match (MR0 - MR3)), this field can either toggle, go low, go high, or do nothing, as selected by EMC1. This field is driven to the MAT pins if the match function is selected via IOPCTL.Writing to this field can directly change the MAT output state. 0b - Low 1b - High
0 EM0	External Match 0 Reflects the state of output MAT0—whether this output is connected to a pin. If a match occurs between Timer Counter (TC) and MR0 (see Match (MR0 - MR3)), this field can either toggle, go low, go high, or

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	do nothing, as selected by EMC0. This field is driven to the MAT pins if the match function is selected via IOPCTL.Writing to this field can directly change the MAT output state. 0b - Low 1b - High

25.7.12 Count Control (CTCR)

Offset

Register	Offset
CTCR	70h

Function

Selects between Timer and Counter mode, and in Counter mode, selects the signal and edges for counting.

When Counter mode is selected as the mode of operation, the CAP input (selected by CINSEL) is sampled on every rising edge of the CTIMER function clock. After comparing two consecutive samples of this CAP input, one of the following four events is recognized:

- Rising edge
- Falling edge
- Either of the edges
- No change in the selected CAP input level

Only if the identified event occurs and the event corresponds to the one selected by CTMODE, Timer Counter (TC) is incremented.

Effective processing of the externally supplied clock to the counter has some limitations:

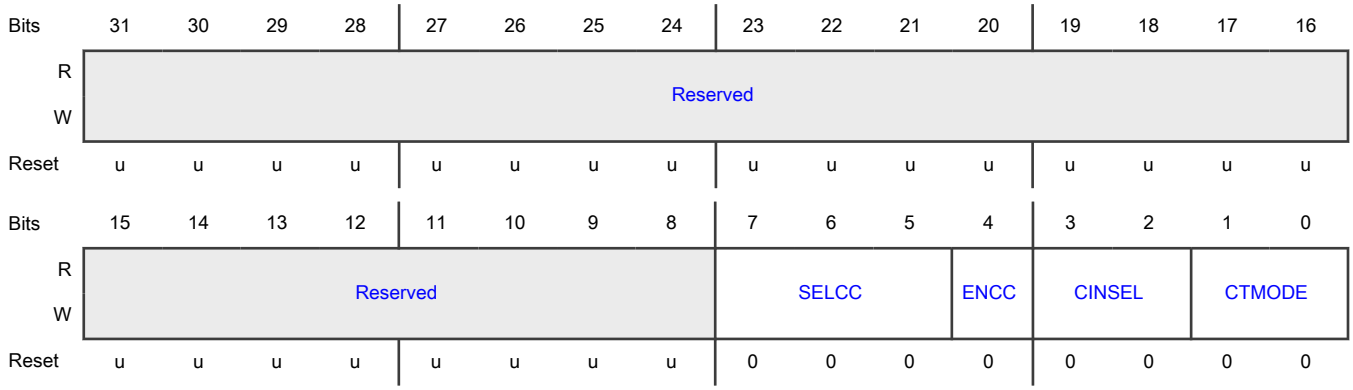
- Because two successive rising edges of the CTIMER function clock are used to identify only one edge on the CAP selected input, the frequency of the CAP input cannot exceed one half of the CTIMER function clock.
- Consequently, the duration of the high or low levels on the same CAP input in this case cannot be shorter than the CTIMER function clock duration.

You can also use ENCC and SELCC to enable and configure the capture-clears-timer feature. This feature allows for a designated edge on a particular CAP input to reset the timer to all zeros. Using this mechanism to clear the timer on the leading edge of an input pulse and perform a capture on the trailing edge permits direct pulse-width measurement using a single capture input without the need to perform a subtraction operation in software.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram



Fields

Field	Function
31-8 —	Reserved
7-5 SELCC	Edge Select Selects which input edge (rising or falling) causes the timer and prescaler to be cleared, when ENCC is 1. These fields have no effect when ENCC is 0. 000b - Capture channel 0 rising edge 001b - Capture channel 0 falling edge 010b - Capture channel 1 rising edge 011b - Capture channel 1 falling edge 100b - Capture channel 2 rising edge 101b - Capture channel 2 falling edge 110b - Capture channel 3 rising edge 111b - Capture channel 3 falling edge
4 ENCC	Capture Channel Enable Enables capture channel. Writing 1 to this field enables clearing of the timer and the prescaler when the capture-edge event specified in SELCC occurs.
3-2 CINSEL	Count Input Select Selects which CAP pin is sampled for clocking when CTMODE is not 0. **NOTE** If Counter mode is selected for a particular CAP*n* input in Count Control (CTCR), you must program the corresponding fields for that input in Capture Control (CCR) as 0. However, you can select both capture and interrupt for the other three CAP*n* inputs in the same timer. 00b - Channel 0, CAPn[0] for CTIMERn

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	01b - Channel 1, CAPn[1] for CTIMERn 10b - Channel 2, CAPn[2] for CTIMERn 11b - Channel 3, CAPn[3] for CTIMERn
1-0 CTMODE	Counter Timer Mode Selects between Timer and Counter mode, and in Counter mode, selects the signal and edges for counting: • If this field is 0, every rising CTIMER function clock edge is incremented. • If this field is 1, Timer Counter (TC) is incremented on the rising edges of the CAP input selected by CINSEL. • If this field is 10, Timer Counter (TC) is incremented on the falling edges of the CAP input selected by CINSEL. • If this field is 11, Timer Counter (TC) is incremented on both edges of the CAP input selected by CINSEL. 00b - Timer mode 01b - Counter mode rising edge 10b - Counter mode falling edge 11b - Counter mode dual edge

25.7.13 PWM Control (PWMC)

Offset

Register	Offset
PWMC	74h

Function

Configures match outputs (see Match mode) as PWM outputs. You can independently set each match output to perform either as a PWM output or match output controlled by External Match (EMR).

NOTE

Different part numbers and package variations may provide different match output pin functions.

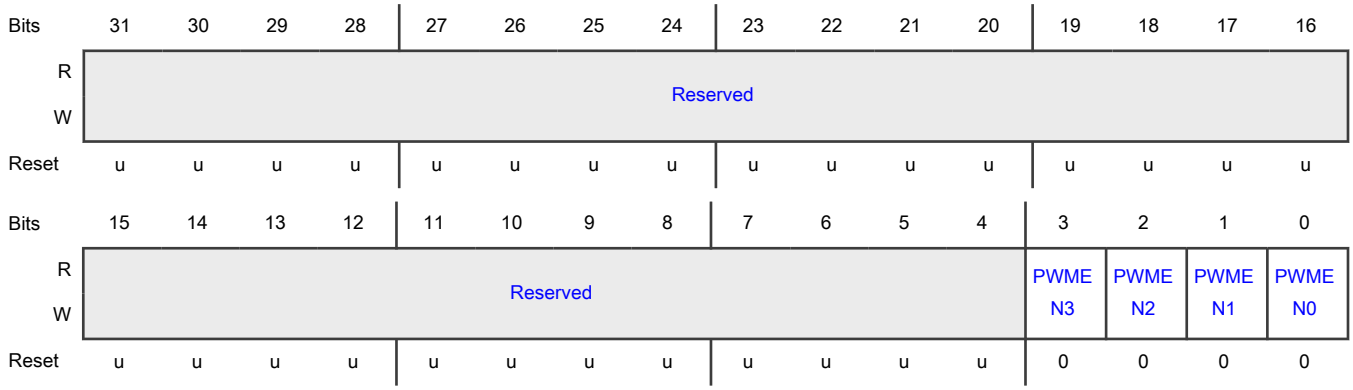
For each timer, you can select a maximum of three single-edge controlled PWM outputs on the MAT*n*[2:0] outputs. One additional match register (see Match (MR0 - MR3)) determines the PWM cycle length. When a match occurs in any of the other match registers, the PWM output is set to high. The match register resets the timer that you can configure to set the PWM cycle length. When the timer is reset to zero, all currently high match outputs configured as PWM outputs are cleared.

This register enables PWM mode for the external match pins.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram



Fields

Field	Function
31-4 —	Reserved
3 PWMEN3	PWM Mode Enable for Channel 3 Enables PWM mode for channel 3. Use match channel 3 to set the PWM cycle length. If this field is 0, EM3 controls CTIMER*n*_MAT3. If this field is 1, PWM mode is enabled for CTIMER*n*_MAT3. 0b - Disable 1b - Enable
2 PWMEN2	PWM Mode Enable for Channel 2 Enables PWM mode for channel 2. If this field is 0, EM2 controls CTIMER*n*_MAT2. If this field is 1, PWM mode is enabled for CTIMER*n*_MAT2. 0b - Disable 1b - Enable
1 PWMEN1	PWM Mode Enable for Channel 1 Enables PWM mode for channel 1. If this field is 0, EM1 controls CTIMER*n*_MAT1. If this field is 1, PWM mode is enabled for CTIMER*n*_MAT1. 0b - Disable 1b - Enable
0 PWMEN0	PWM Mode Enable for Channel 0 Enables PWM mode for channel 0. If this field is 0, EM0 controls CTIMER*n*_MAT0. If this field is 1, PWM mode is enabled for CTIMER*n*_MAT0. 0b - Disable 1b - Enable

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

25.7.14 Match Shadow (MSR0 - MSR3)

Offset

Register	Offset
MSR0	78h
MSR1	7Ch
MSR2	80h
MSR3	84h

Function

Contains the value that the corresponding Match (MR0 - MR3) are (optionally) reloaded with at the start of each new counter cycle. Typically, the match that causes the counter to be reset (and instigates the match reload) is also programmed to generate an interrupt or DMA request. Software or the DMA engine then has one full counter cycle to modify the contents of this register before the next reload occurs.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	MATCH_SHADOW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MATCH_SHADOW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-0 MATCH_SHADOW	Timer Counter Match Shadow Value Reloads automatically with the contents of this register whenever Timer Counter (TC) is reset to zero.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Chapter 26
Windowed Watchdog Timer (WWDT)

26.1 Chip-specific WWDT information

Table 155. Reference links to related information

Topic	Related module	Reference
Full description	WWDT	WWDT
System memory map		Memory map
Clocking		Clock distribution
Power Management		Power Management
Signal multiplexing	Port control	Signal multiplexing

26.1.1 Module instances

This chip has two instances of the WWDT module, WWDT0 and WWDT1.

26.1.2 LOCK function

WWDT supports lock function. Once WWDT is locked, WWDT configuration, including clock, can’t be changed.

26.2 Overview

WWDT helps you reset or interrupt a microcontroller within a programmable time, if the microcontroller (or core) is stuck in an infinite loop or is executing an unintended code. If an application fails to reload or feed a watchdog timer within the predefined duration of time, it generates a watchdog reset (if enabled).

26.2.1 Block diagram

In the following WWDT block diagram, the synchronization logic (APB bus clock to WDCLK) is not shown. See the chip-specific WWDT information section or clocking chapter for more details.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

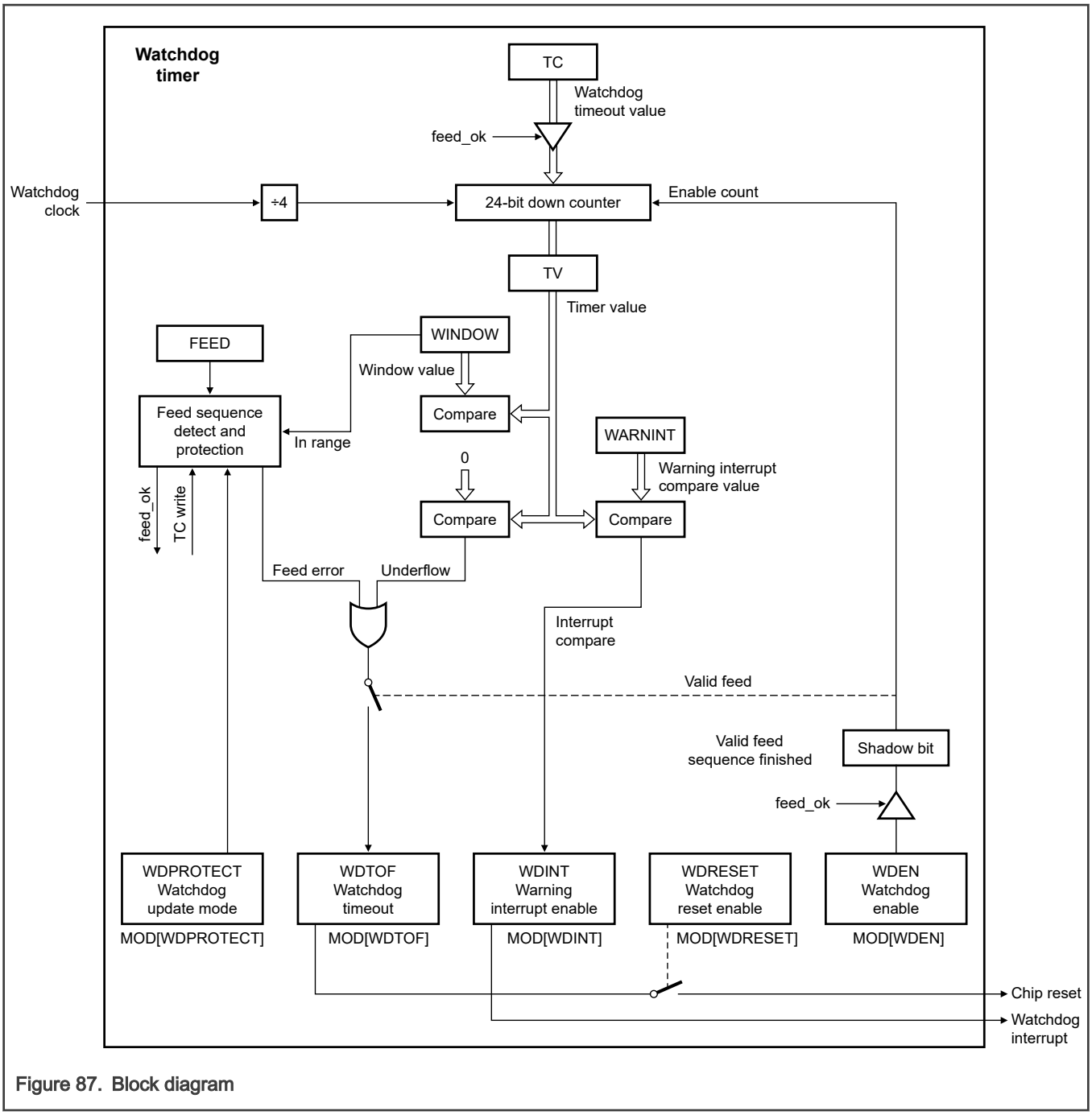


Figure 87. Block diagram

26.2.2 Features

- Can reset or interrupt the microcontroller within a programmable time
- Watchdog reset indication flag
- Clocking features:
  - Programmable 24-bit timer with an internal fixed (divided by 4) prescaler
  - Selectable time period in increments of four watchdog clocks: From 1024 watchdog clocks (T_{WDCLK} x 256 x 4) to over 67 million watchdog clocks (T_{WDCLK} x 2^{24} x 4)

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

— Watchdog clock (WDCLK) source having a selectable frequency in the range of 6 kHz to 1.5 MHz (see the chip-specific information section and the Clocking chapter for detailed setting in this device)

- Can be configured to run in Low-Power mode (see the chip-specific information section and the Power Management chapter for detailed setting in this device)
- Other configurable features:
  - — Windowed operation requiring a watchdog reload to occur between a programmable minimum and maximum timeout period
  - — Generation of a warning interrupt at a programmable time before watchdog timeout
  - — Protection of the watchdog reload value so that you can change it only after the warning interrupt time is reached

26.3 Functional description

When you program a watchdog window (a timing window), an early watchdog reload is also treated as a watchdog event, which prevents situations where a system failure may still reload the watchdog timer. For example, the application code could be stuck in an interrupt service that contains a watchdog reload (feed). You must set the timing window in a way that this situation causes an early watchdog timer reload, which generates a watchdog event, and then enables the system to recover from this situation.

WWDT consists of a fixed (divided by 4) prescaler and a 24-bit counter. The minimum value of the counter is FFh; if you write a value lower than FFh, the counter automatically loads with the value FFh.

- Minimum watchdog interval is (T_{WDCLK} X 256 X 4).
- Maximum watchdog interval is (T_{WDCLK} X 2^{24} X 4) in multiples of (T_{WDCLK} X 4).

26.3.1 Operating modes

This section describes all functional operating modes of the WWDT module: Disabled, Watchdog Interrupt, and Watchdog Reset.

Table 156. Watchdog operating modes selection

MOD[WDEN]	MOD[WDRESET]	Mode	Mode description
0	X (0 or 1)	Disabled	Debugs or operates without the watchdog running.
1	0	Watchdog Interrupt	Generates a watchdog warning interrupt but does not enable the watchdog reset. A watchdog counter equal to the value of WARNINT[WARNINT] sets MOD[WDINT] and generates a watchdog interrupt request.
1	1	Watchdog Reset	Enables both the watchdog interrupt and watchdog reset: • A watchdog counter equal to the value of WARNINT[WARNINT] sets MOD[WDINT] and generates a watchdog interrupt request. • A watchdog counter equal to 0 resets the microcontroller. • A watchdog feed before reaching the value of WINDOW[WINDOW] also causes a watchdog reset.

26.3.2 Example timing diagrams to show WWDT in operation

A feed is correct when both of the following conditions are true:

- A valid feed sequence completes, writing AAh followed by 55h to FEED[FEED].
- The value of TV[COUNT] is not greater than the value of WINDOW[WINDOW].

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

If either of these conditions is not true, a feed error occurs.

In the correct watchdog feed with Windowed mode enabled, the sequence of writing AAh followed by 55h occurs when the counter value (11FCh) is not greater than the value of WINDOW[WINDOW] (1200h). It is a correct feed (reload) event. After the correct feed event occurs, the watchdog counter reloads with the TC[COUNT] value (2000h).

In addition to the above two conditions, when MOD[WDPROTECT] = 1, the watchdog counter must not be greater than the value of WARNINT[WARNINT] for a correct feed to occur. See the following figure for a pictorial representation.

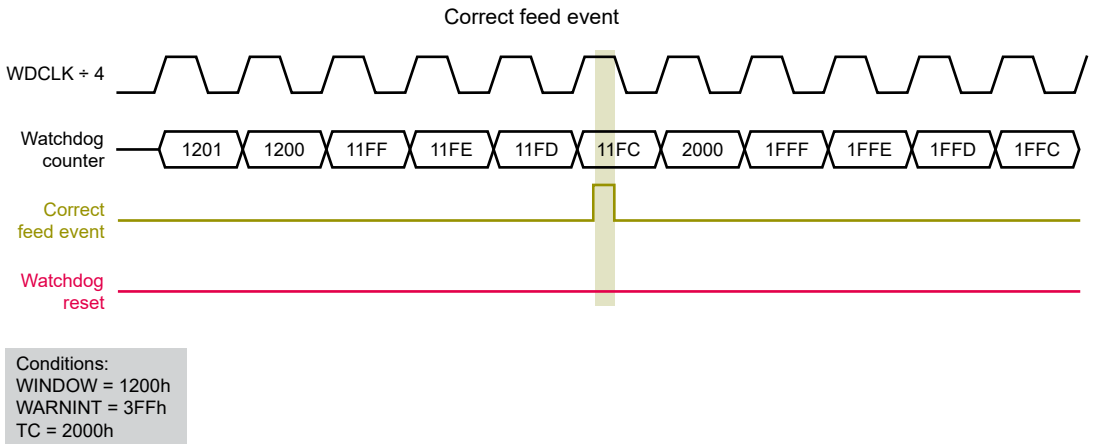


Figure 88. Correct watchdog feed with Windowed mode enabled

A correct feed sequence can only occur when the value of TV[COUNT] is not greater than the value of WINDOW[WINDOW] (1200h). Otherwise, a feed error occurs.

The following figure shows a reset trigger when you feed the watchdog too early.

An early feed event occurs if the feed sequence occurs when the watchdog counter value (1257h) is greater than the value of WINDOW[WINDOW] (1200h). This feed error can generate a reset if MOD[WDRESET] = 1.

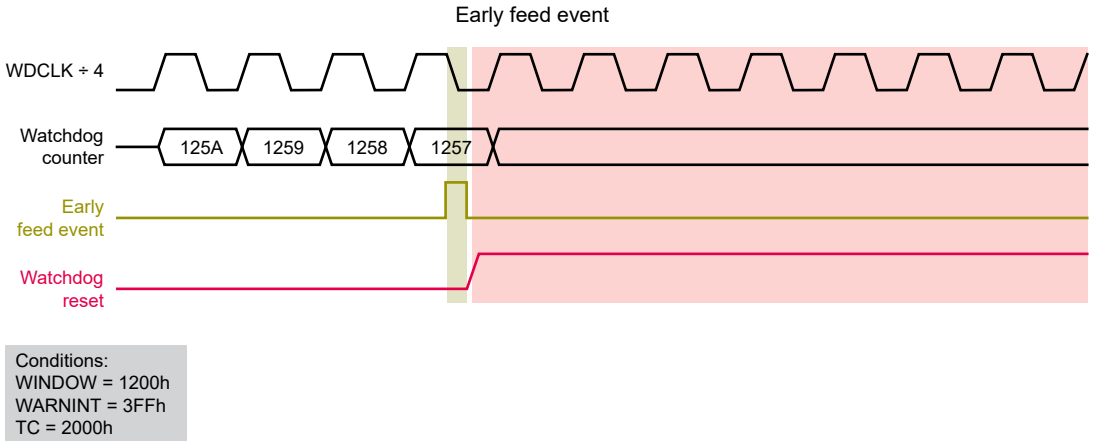


Figure 89. Early watchdog feed with Windowed mode enabled

The following figure shows WWDT generating a warning.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

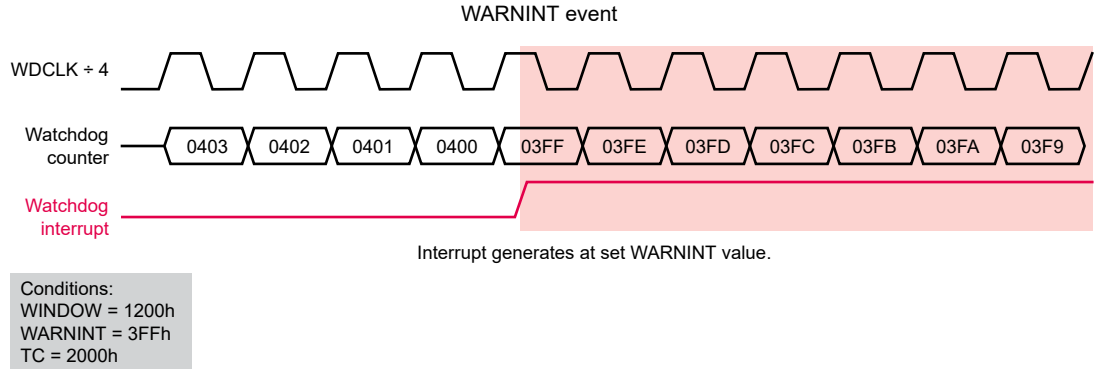


Figure 90. Watchdog warning interrupt

26.3.3 Clocking

WWDT uses two clocks: APB bus clock and WDCLK.

- The system clock derives the APB bus clock that is used for APB accesses to the watchdog registers.
- The watchdog oscillator derives WDCLK that is used for watchdog timer counting.

The synchronization logic between the two clock domains works as follows:

- When updating Mode (MOD) and Timer Constant (TC), the new value takes effect in three WDCLK cycles on the logic in the WDCLK clock domain.
- When the watchdog timer is counting on WDCLK, the synchronization logic first locks the value of the counter on WDCLK and then synchronizes it with the APB bus clock, so that the CPU can read Timer Value (TV).

NOTE

Because of the synchronization step, you must add a delay of three WDCLK clock cycles between the feed sequence and the time that you take to enable the functionality of MOD[WDPROTECT]. The length of the delay depends on the selected watchdog clock, WDCLK.

26.3.4 Using the WWDT lock

WWDT supports lock features to ensure continuous operation. These features prevent:

- The disabling of the WWDT clock source.
- The changing of the WWDT reload value.

26.3.4.1 Preventing the disabling of the WWDT clock source

If MOD[LOCK] = 1, the WWDT clock source is locked. You cannot disable the clock source when entering Sleep or Deep-Sleep modes. Therefore, enable the watchdog oscillator for each Power mode before writing 1 to MOD[LOCK].

26.3.4.2 Preventing the changing of the WWDT reload value

If MOD[WDPROTECT] = 1, then any attempt to change the value of TC[COUNT] with a feed sequence, before the watchdog counter value becomes less than the values of WARNINT[WARNINT] and WINDOW[WINDOW], sets MOD[WDTOF] and causes a watchdog reset.

Any type of reset disables the reload overwrite lock mechanism.

26.4 External signals

This module has no external signals.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

26.5 Initialization

Perform the following steps to initialize WWDT:

1. Enable and configure the watchdog oscillator.
2. Set the watchdog timer constant reload value using TC[COUNT].
3. Set the watchdog timer operating mode using MOD[WDPROTECT].
4. Set a value for the watchdog window time using WINDOW[WINDOW], if the windowed operation is desired.
5. Set a value for the watchdog warning interrupt using WARNINT[WARNINT] and configure the system interrupt controller, if a warning interrupt is desired.
6. Enable the watchdog by writing AAh followed by 55h to FEED[FEED].

To prevent a watchdog event, you must feed or reload WWDT again before the watchdog counter reaches 0. If you set a value for the watchdog window time, then the feed or reload must also occur after the watchdog counter passes the value of WINDOW[WINDOW].

After you configure WWDT in a way that a watchdog event causes a reset (MOD[WDRESET] = 1), or when the counter reaches 0 (it causes a reset), then the microcontroller (or core) is also reset (loading the stack pointer and program counter from the vector table, just like the way it happens for an external reset).

26.6 Memory map and register definition

This section includes the WWDT module memory map and detailed descriptions of all registers.

26.6.1 WWDT register descriptions

26.6.1.1 WWDT memory map

WWDT0 base address: 4000_C000h

WWDT1 base address: 4000_D000h

Offset	Register	Width (In bits)	Access	Reset value
0h	Mode (MOD)	32	RW	0000_0000h
4h	Timer Constant (TC)	32	RW	0000_00FFh
8h	Feed Sequence (FEED)	32	W	See section
Ch	Timer Value (TV)	32	R	0000_00FFh
14h	Warning Interrupt Compare Value (WARNINT)	32	RW	0000_0000h
18h	Window Compare Value (WINDOW)	32	RW	00FF_FFFFh

26.6.1.2 Mode (MOD)

Offset

Register	Offset
MOD	0h

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Function

Controls the operation of WWDT. You must perform a watchdog feed before any changes to this register take effect.

NOTE

After you write 1 to MOD[WDEN], MOD[WDPROTECT], or MOD[WDRESET], you can return those fields to 0 only with an external reset or a WWDT reset. Until those resets occur, writing 0 to the aforementioned fields has no effect.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0									Reserved	LOCK	WDPROTE...	WDINT	WDTOF	WDRESET	WDEN
W													W1C			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-7 —	Reserved
6 —	Reserved
5 LOCK	Lock Prevents disabling or powering down of the watchdog oscillator after you write 1 to this field and perform a watchdog feed. You can write 1 to this field, which can become 0 only with a reset. 0b - No Lock 1b - Lock
4 WDPROTECT	Watchdog Update Mode Protects the value of TC[COUNT] from changing at any time. In Flexible mode (when this field is 0), you can change the value of TC[COUNT] at any time. In Threshold mode (when this field is 1), you can change the value of TC[COUNT] only after the counter is less than the individual values of WARNINT[WARNINT] and WINDOW[WINDOW]. Any attempt to change the value of TC[COUNT] with a feed sequence, before the watchdog counter value becomes less than the values of WARNINT[WARNINT] and WINDOW[WINDOW], sets MOD[WDTOF] and causes a watchdog reset. You can write 1 to this field, which can become 0 only with a reset (see the note in MOD register).

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	0b - Flexible 1b - Threshold
3 WDINT	Warning Interrupt Flag Sets the flag when the timer is at or below the value defined in WARNINT[WARNINT]. Note that you cannot clear this flag while the value of WARNINT[WARNINT] is equal to the value of TV[COUNT]. This can occur if the value of both WARNINT[WARNINT] and MOD[WDRESET] is 0 when TV[COUNT] decrements to 0. When you configure WWDT to generate a warning interrupt, the interrupt occurs when the counter is no longer greater than the value defined by WARNINT[WARNINT]. If a feed sequence occurs when the watchdog counter value is greater than the value of WINDOW[WINDOW], this flag is set. Any reset clears this flag. 0b - No flag 1b - Flag
2 WDTOF	Watchdog Timeout Flag The bit could be set if: • Watchdog timer timeout has occurred • Feed error has occurred You must clear this flag by writing 0 to it. An external reset or POR clears this flag too. 0b - Watchdog event has not occurred. 1b - Watchdog event has occurred (causes a chip reset if WDRESET = 1).
1 WDRESET	Watchdog Reset Enable Allows a watchdog timeout to cause a chip reset, if this field = 1. After you write 1 to this field, you cannot write 0 to it (see the note in MOD register). **NOTE** If you write 0 to this field, and after an interrupt occurs, do not write 1 to this field immediately unless you refresh the watchdog. 0b - Interrupt 1b - Reset
0 WDEN	Watchdog Enable Enables or disables the watchdog timer. After writing 1 to this field, perform a watchdog feed (reload) to enable the watchdog timer. After you write 1 to this field, you cannot write 0 to it (see the note in MOD register). 0b - Timer stopped

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	1b - Timer running

26.6.1.3 Timer Constant (TC)

Offset

Register	Offset
TC	4h

Function

Specifies the desired timeout value. Transferring the value of this register into the watchdog counter requires a feed sequence. The register resets to 00_00FFh. Writing a value below FFh causes 00_00FFh to load into the register. Therefore, the minimum timeout interval is T_{WDCLK} X 256 X 4.

If MOD[WDPROTECT] = 1, then any attempt to change the value of this register with a feed sequence, before the watchdog counter value becomes less than the individual values of WARNINT[WARNINT] and WINDOW[WINDOW], sets MOD[WDTOF] and causes a watchdog reset.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0								COUNT							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COUNT															
W																
Reset	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

Fields

Field	Function
31-24 —	Reserved
23-0 COUNT	Watchdog Timeout Value Specifies the desired timeout value for WWDT.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

26.6.1.4 Feed Sequence (FEED)

Offset

Register	Offset
FEED	8h

Function

Reloads the watchdog timer with the TC[COUNT] value, if you write AAh followed by 55h to FEED[FEED]. This operation also starts WWDT, if the watchdog timer is enabled by writing 1 to MOD[WDEN].

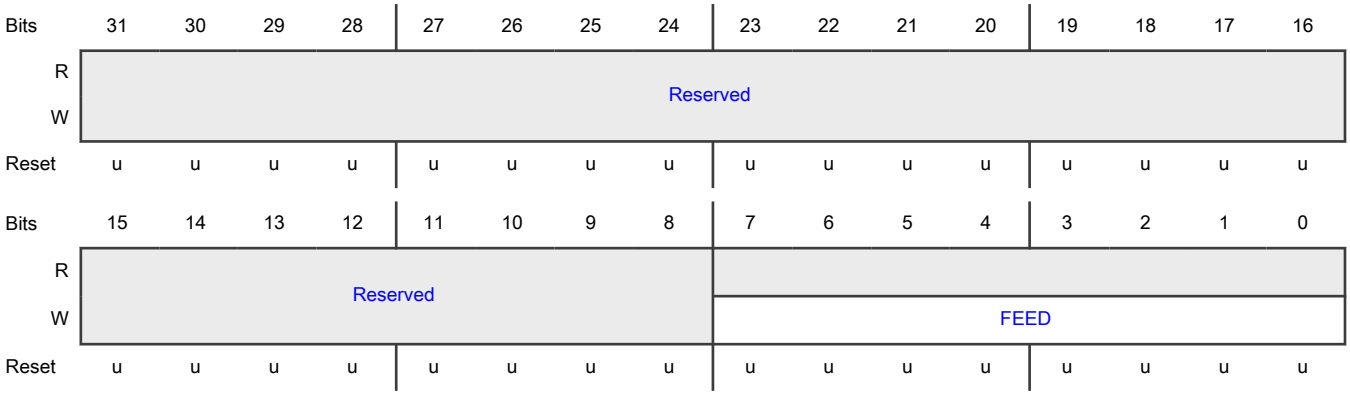
However, writing 1 to MOD[WDEN] is not sufficient to enable WWDT. Before the watchdog can generate a reset, MOD[WDEN] must be 1, followed by a valid feed sequence that completes itself. Until then, the watchdog ignores feed errors.

After writing AAh to FEED[FEED], access to any watchdog register other than writing 55h to FEED[FEED] causes an immediate reset or interrupt when WWDT is enabled and sets MOD[WDTOF]. An incorrect access to a watchdog register during a feed sequence generates the reset during the second APB bus clock.

To avoid an unintended interrupt, it is a good practice to disable interrupts around a feed sequence. Disable interrupts if they may result in rescheduling processor control away from the current task in the middle of the feed, and then lead to some other access to WWDT before control returns to the interrupted task.

If the value of WINDOW[WINDOW] is less than the default value, it may limit the time for which a watchdog feed is allowed.

Diagram



Fields

Field	Function
31-8 —	Reserved
7-0 FEED	Feed Value Specifies feed value. The value must be AAh followed by 55h.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

26.6.1.5 Timer Value (TV)

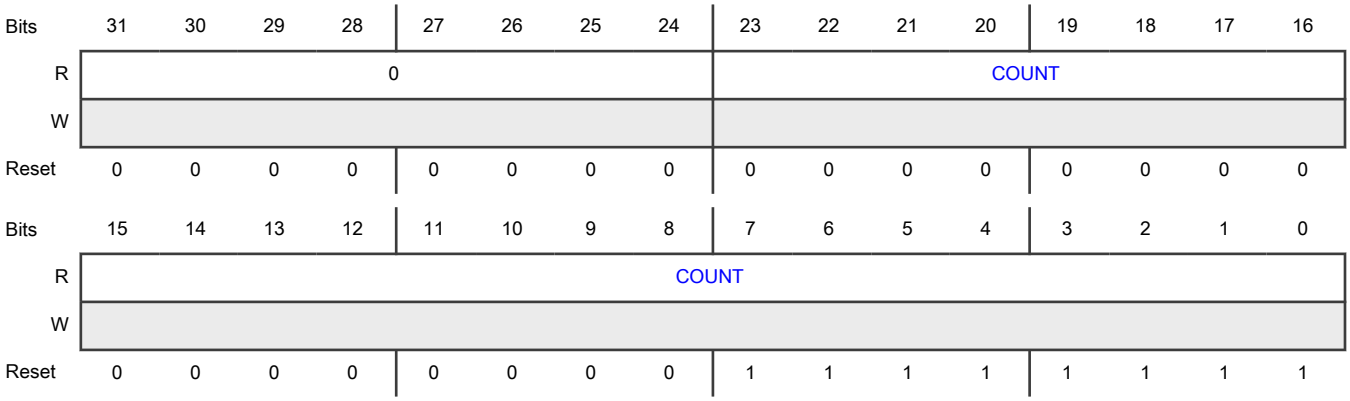
Offset

Register	Offset
TV	Ch

Function

Contains the current value of the WWDT counter. When the core (or CPU) reads this register, the value of this register is older than the actual value of the timer. Reading the timer using the lock and synchronization procedure takes up to six WDCLK cycles plus six APB bus clock cycles.

Diagram



Fields

Field	Function
31-24 —	Reserved
23-0 COUNT	Counter Timer Value Specifies the current value of the watchdog timer.

26.6.1.6 Warning Interrupt Compare Value (WARNINT)

Offset

Register	Offset
WARNINT	14h

Function

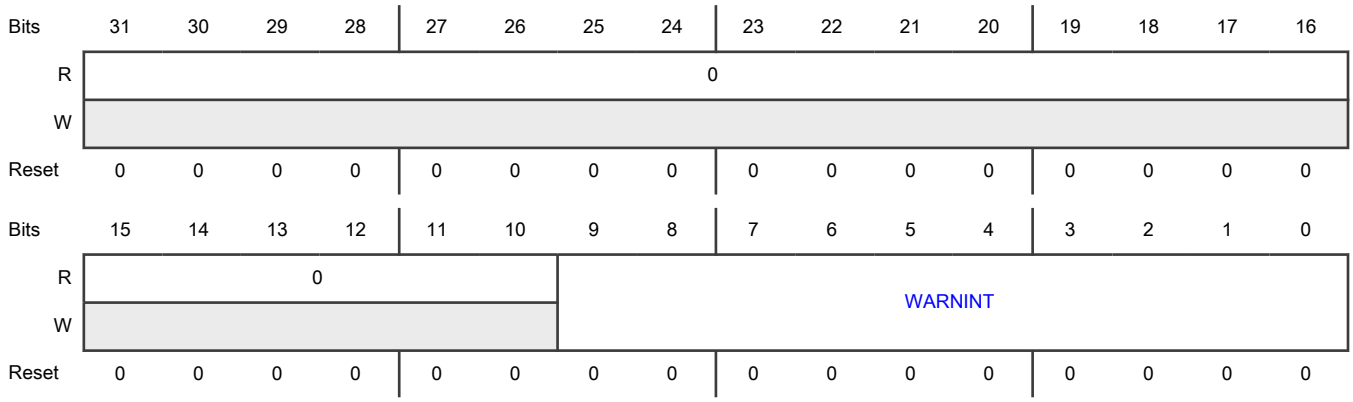
Determines the value of TV[COUNT] that generates a watchdog interrupt. When this value is less than or equal to the value defined by WARNINT, an interrupt generates after the subsequent WDCLK.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

A match of the WWDT counter to WARNINT occurs when the lower 10 bits of the timer counter have the same value as the lower 10 bits of WARNINT, and the remaining upper bits of the counter are all 0. This gives a maximum time of 1023 watchdog timer counts (4096 watchdog clocks) for the interrupt to occur before a watchdog event. If WARNINT is 0, then the interrupt occurs at the same time as the watchdog event.

Diagram



Fields

Field	Function
31-10 —	Reserved
9-0 WARNINT	Watchdog Warning Interrupt Compare Value Specifies a timer value that generates a warning interrupt.

26.6.1.7 Window Compare Value (WINDOW)

Offset

Register	Offset
WINDOW	18h

Function

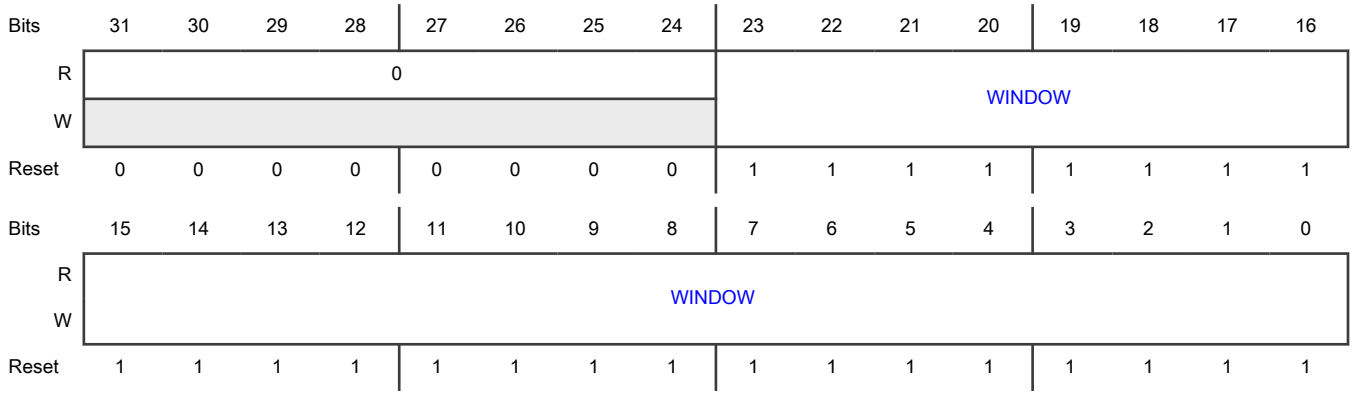
Determines the highest timer value (TV[COUNT]) allowed during a watchdog feed. If a feed sequence occurs when the value of TV[COUNT] is greater than the value in WINDOW[WINDOW], a watchdog event occurs.

WINDOW[WINDOW] resets to the maximum value of TV[COUNT], where windowing is not in effect.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram



Fields

Field	Function
31-24 —	Reserved
23-0 WINDOW	Watchdog Window Value Specifies the highest timer value in which a watchdog feed can occur.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Chapter 27
Real Time Clock (RTC)

27.1 Chip-specific RTC information

Table 157. Reference links to related information

Topic	Related module	Reference
Full description	RTC	RTC
System memory map		Memory map
Clocking		Clock distribution
Power Management		Power Management
Signal multiplexing	Port control	Signal multiplexing

27.1.1 RTC clock

FRO is FRO16K from VBAT, and OSC32k is available in this chip.

27.2 Overview

27.2.1 Block diagram

The following figure is the block diagram of this module.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

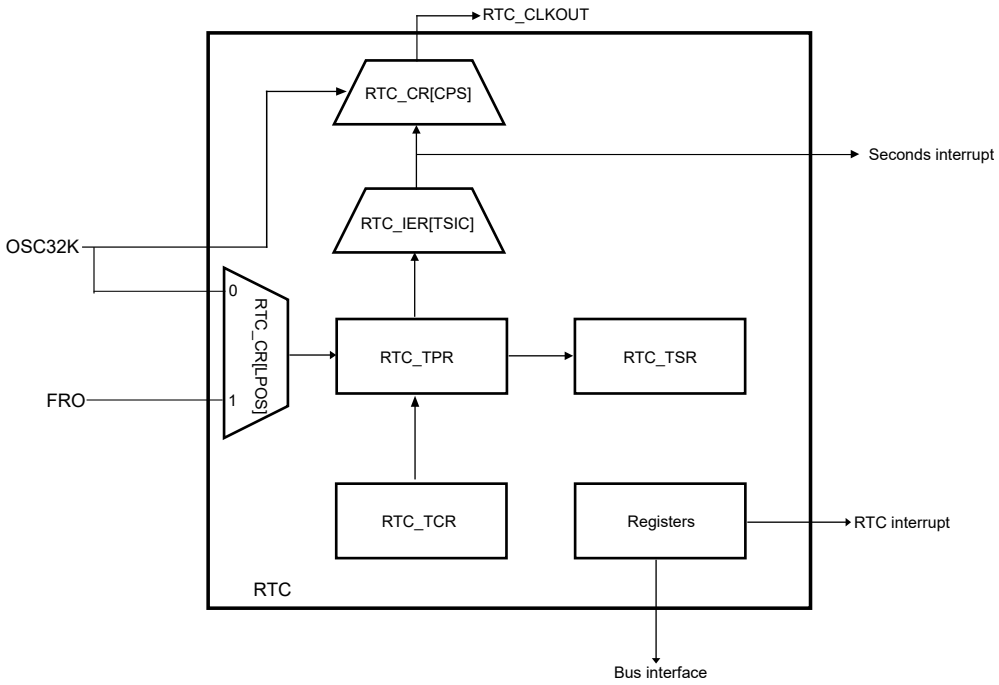


Figure 91. RTC module block diagram

27.2.2 Features

The RTC module features include:

- 32-bit seconds counter with roll-over protection and 32-bit alarm
- 16-bit prescaler with compensation that can correct errors between 0.12 ppm and 3906 ppm
- Option to increment prescaler using a 1 kHz LPO (prescaler increments by 32 every clock edge)
- Register write protection
  - Lock register requires POR or software reset to enable write access
- Configurable 1, 2, 4, 8, 16, 32, 64 or 128 Hz square wave output with optional interrupt

27.3 Functional description

27.3.1 Power

The RTC is an always powered block that remains active in all low power modes.

27.3.2 Time counter

The time counter consists of a 32-bit seconds counter that increments once every second and a 16-bit prescaler register that increments once every 32.768 kHz clock cycle. There is also the option to clock the prescaler using a 1 kHz LPO that increments the prescaler by 32 on every clock cycle.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Reading the time counter (either seconds or prescaler) while it is incrementing may return invalid data due to synchronization of the read data bus. If it is necessary for software to read the prescaler or seconds counter when they could be incrementing, it is recommended that two read accesses are performed and that software verifies that the same data was returned for both reads.

The time seconds register and time prescaler register can be written only when SR[TCE] is clear. Always write to the prescaler register before writing to the seconds register, because the seconds register increments on the falling edge of bit 14 of the prescaler register.

The time prescaler register increments provided SR[TCE] is set, SR[TIF] is clear, SR[TOF] is clear, and the 32.768 kHz (or 1 kHz) clock source is present. After enabling the oscillator, wait the oscillator startup time before setting SR[TCE] to allow time for the oscillator clock output to stabilize.

If the time seconds register overflows, SR[TOF] is set and the time prescaler register stops incrementing. Clear SR[TOF] by initializing the time seconds register. The time seconds register and time prescaler register read as zero whenever SR[TOF] is set.

SR[TIF] is set on POR and software reset and is cleared by initializing the time seconds register. The time seconds register and time prescaler register read as zero whenever SR[TIF] is set.

27.3.3 Compensation

The compensation logic provides an accurate and wide compensation range and can correct errors as high as 3906 ppm and as low as 0.12 ppm. The compensation factor must be calculated externally to the RTC and supplied by software to the compensation register. The RTC itself does not calculate the amount of compensation that is required, although the 1 Hz clock is output to an external pin in support of external calibration logic.

Crystal compensation can be supported by using firmware and crystal characteristics to determine the compensation amount. Temperature compensation can be supported by firmware that periodically measures the external temperature via ADC and updates the compensation register based on a look-up table that specifies the change in crystal frequency over temperature.

The compensation logic alters the number of 32.768 kHz clock cycles it takes for the prescaler register to overflow and increment the time seconds counter. The time compensation value is used to adjust the number of clock cycles between -127 and +128. Cycles are added or subtracted from the prescaler register when the prescaler register equals 0x3FFF and then increments. The compensation interval is used to adjust the frequency at which the time compensation value is used, that is, from once a second to once every 256 seconds.

Updates to the time compensation register do not take effect until the next time the time seconds register increments and provided the previous compensation interval has expired. When the compensation interval is set to other than once a second then the compensation is applied in the first second interval and the remaining second intervals receive no compensation.

Compensation is disabled by configuring the time compensation register to zero.

When the prescaler is configured to increment using the 1 kHz LPO, the effective compensation value is divided by 32 and can only adjust the number of clock cycles between -4 and +3.

27.3.4 Time alarm

The Time Alarm register (TAR), SR[TAF], and IER[TAIE] allow the RTC to generate an interrupt at a predefined time. The 32-bit TAR is compared with the 32-bit Time Seconds register (TSR) each time it increments. SR[TAF] is set when TAR equals TSR and TSR increments.

SR[TAF] is cleared by writing TAR. This is usually the next alarm value, although writing a value that is less than TSR (such as 0) prevents SR[TAF] from setting again. SR[TAF] cannot otherwise be disabled, although the interrupt it generates is enabled or disabled by IER[TAIE].

27.3.5 Update mode

The Update Mode field in the Control register (CR[UM]) configures software write access to the Time Counter Enable (SR[TCE]) field. When CR[UM] is clear, SR[TCE] can be written only when LR[SRL] is set. When CR[UM] is set, SR[TCE] can also be written when SR[TCE] is clear or when SR[TIF] or SR[TOF] are set. This allows the time seconds and prescaler registers to be initialized whenever time is invalidated, while preventing the time seconds and prescaler registers from being changed on the fly. When LR[SRL] is set, CR[UM] has no effect on SR[TCE].

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

27.3.6 Register lock

The Lock register (LR) can be used to block write accesses to certain registers until the next POR or software reset. Locking the Control register (CR) disables the software reset. Locking LR blocks future updates to LR.

Write accesses to a locked register are ignored and do not generate a bus error.

27.3.7 Modes of operation

The RTC remains functional in all low power modes and can generate an interrupt to exit any low power mode.

27.3.8 Clocking

The time counter within the RTC is clocked by default from a 32.768 kHz clock. Alternatively, the time counter can be clocked by a LPO 1 kHz clock and the prescaler increments by 32 for each LPO clock.

The 32.768 kHz crystal oscillator is disabled at POR and must be enabled by software. After enabling the crystal oscillator, wait the oscillator startup time before setting SR[TCE] or using the oscillator clock external to the RTC.

The crystal oscillator includes tunable capacitors that can be configured by software. Do not change the capacitance unless the oscillator is disabled.

27.3.9 Reset

The power-on-reset signal initializes all RTC registers to their default state. A software reset bit can also initialize all RTC registers.

Writing 1 to CR[SWR] forces the equivalent of a POR to the rest of the RTC module. CR[SWR] is not affected by the software reset and must be cleared by software.

27.3.10 Interrupts

The RTC interrupt is asserted whenever a status flag and the corresponding interrupt enable bit are both set. It is always asserted on POR, and software reset. The RTC interrupt is enabled at the chip level by enabling the chip-specific RTC clock gate control bit. The RTC interrupt can be used to wakeup the chip from any low-power mode.

The RTC seconds interrupt is an edge-sensitive interrupt with a dedicated interrupt vector that is generated once a second and requires no software overhead (there is no corresponding status flag to clear). It is enabled in the RTC by the time seconds interrupt enable bit and enabled at the chip level by setting the chip-specific RTC clock gate control bit. The frequency of the seconds interrupt defaults to 1 Hz, but can instead be configured to trigger every 2, 4, 8, 16, 32, 64 or 128 Hz.

27.4 Initialization

To initialize the RTC time counter:

1. Enable the desired clock source for the RTC and configure RTC_CR[LPOS] to select that clock.
2. Initialize the time prescaler register by writing RTC_TPR[TPR] with the appropriate time value between 0x0 and 0x7FFF.
3. Initialize the time second register by writing RTC_TSR[TSR] with the appropriate time value that is greater than 0x0.
4. Enable the time counter by writing RTC_SR[SR]=0x10.

27.5 Register descriptions

All registers must be accessed using 32-bit writes and all register accesses incur three wait states.

When the supervisor access control bit is clear, only supervisor mode software can write to the RTC registers. Writes to the RTC registers by non-supervisor mode software generate a bus error. Both supervisor and non-supervisor mode software can always read the RTC registers.

Writing to a register protected by the lock register does not generate a bus error, but the writes can not be completed.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

27.5.1 RTC register descriptions

27.5.1.1 RTC memory map

RTC0 base address: 400E_E000h

Offset	Register	Width (In bits)	Access	Reset value
0h	RTC Time Seconds (TSR)	32	RW	0000_0000h
4h	RTC Time Prescaler (TPR)	32	RW	0000_0000h
8h	RTC Time Alarm (TAR)	32	RW	0000_0000h
Ch	RTC Time Compensation (TCR)	32	RW	0000_0000h
10h	RTC Control (CR)	32	RW	0000_0080h
14h	RTC Status (SR)	32	RW	0000_0001h
18h	RTC Lock (LR)	32	RW	0000_00FFh
1Ch	RTC Interrupt Enable (IER)	32	RW	0000_0007h

27.5.1.2 RTC Time Seconds (TSR)

Offset

Register	Offset
TSR	0h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	TSR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	TSR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-0	Time Seconds Register

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

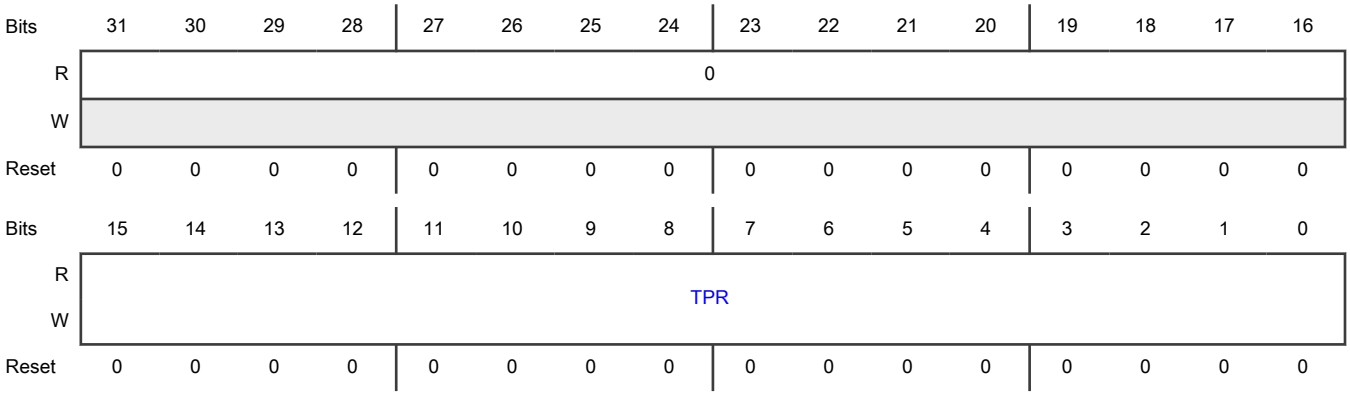
Field	Function
TSR	When the time counter is enabled, the TSR is read only and increments once a second provided SR[TOF] or SR[TIF] is not set. The time counter reads as zero when SR[TOF] or SR[TIF] is set. When the time counter is disabled, the TSR can be read or written. Writing to the TSR when the time counter is disabled clears SR[TOF] or SR[TIF]. Writing to TSR with zero is supported, but not recommended because TSR reads as zero when SR[TIF] or SR[TOF] is set (indicating the time is invalid).

27.5.1.3 RTC Time Prescaler (TPR)

Offset

Register	Offset
TPR	4h

Diagram



Fields

Field	Function
31-16 —	Reserved
15-0 TPR	Time Prescaler Register When the time counter is enabled, the TPR is read only and increments every 32.768 kHz clock cycle. The time counter reads as zero when SR[TOF] or SR[TIF] are set. When the time counter is disabled, the TPR can be read or written. The TSR[TSR] increments when bit 14 of the TPR transitions from a logic one to a logic zero.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

27.5.1.4 RTC Time Alarm (TAR)

Offset

Register	Offset
TAR	8h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	TAR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	TAR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-0 TAR	Time Alarm Register When the time counter is enabled, the SR[TAF] is set whenever the TAR[TAR] equals the TSR[TSR] and the TSR[TSR] increments. Writing to the TAR clears the SR[TAF].

27.5.1.5 RTC Time Compensation (TCR)

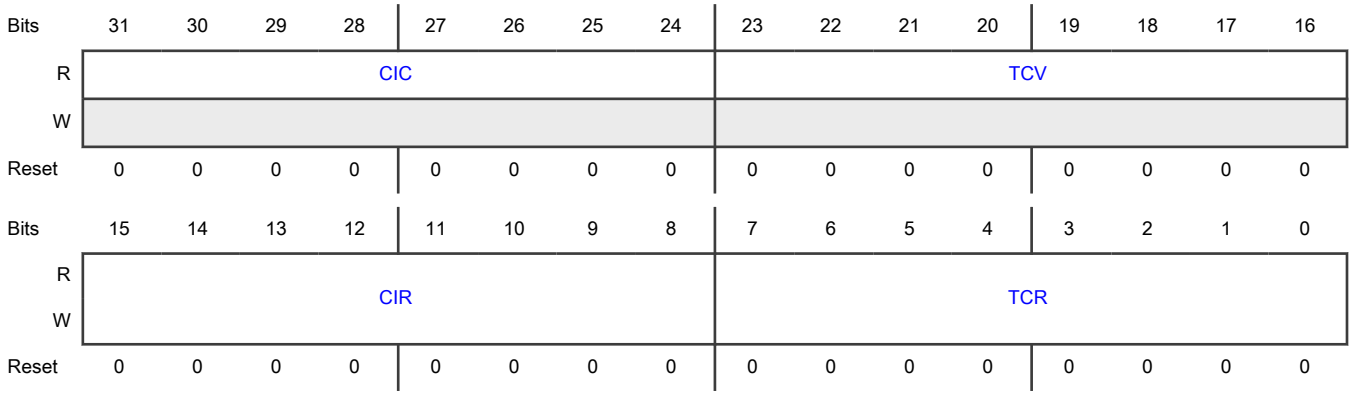
Offset

Register	Offset
TCR	Ch

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram



Fields

Field	Function
31-24 CIC	Compensation Interval Counter Current value of the compensation interval counter. If the compensation interval counter equals zero then it is loaded with the contents of the CIR. If the CIC does not equal zero then it is decremented once a second. Reading this register at the same time as the seconds counter is incrementing can result in an incorrect value being read.
23-16 TCV	Time Compensation Value Current value used by the compensation logic for the present second interval. Updated once a second if the CIC equals 0 with the contents of the TCR field. If the CIC does not equal zero then it is loaded with zero (compensation is not enabled for that second increment). Reading this register at the same time as the seconds counter is incrementing can result in an incorrect value being read.
15-8 CIR	Compensation Interval Register Configures the compensation interval in seconds from 1 to 256 to control how frequently the TCR should adjust the number of 32.768 kHz cycles in each second. The value written should be one less than the number of seconds. For example, write zero to configure for a compensation interval of one second. This register is double buffered and writes do not take effect until the end of the current compensation interval.
7-0 TCR	Time Compensation Register Configures the number of 32.768 kHz clock cycles in each second, equal to 32,768 - TCR (where TCR is a twos complement sign extended value). This register is double buffered and writes do not take effect until the end of the current compensation interval. Some example values are shown below. 0000_0000b - Time Prescaler Register overflows every 32768 clock cycles. 0000_0001b - Time Prescaler Register overflows every 32767 clock cycles. 0111_1110b - Time Prescaler Register overflows every 32642 clock cycles. 0111_1111b - Time Prescaler Register overflows every 32641 clock cycles. 1000_0000b - Time Prescaler Register overflows every 32896 clock cycles. 1000_0001b - Time Prescaler Register overflows every 32895 clock cycles. 1111_1111b - Time Prescaler Register overflows every 32769 clock cycles.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

27.5.1.6 RTC Control (CR)

Offset

Register	Offset
CR	10h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved								0						Reserved	
W	0														0	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	LPOS	0	Reserved	Reserved	UM	Reserved	Reserved	SWR
W	0	0	0	0	0	0	0	0			0	0		0	0	
Reset	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0

Fields

Field	Function
31-24 —	Reserved
23-18 —	Reserved
17-16 —	Reserved
15 —	Reserved
14 —	Reserved
13 —	Reserved
12 —	Reserved

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
11 —	Reserved
10 —	Reserved
9 —	Reserved
8 —	Reserved
7 LPOS	LPO Select When set, the RTC prescaler increments using an internal clock. The LPO increments the prescaler from bit TPR[5] (TPR[4:0] are ignored), supporting close to 1 second increment of the seconds register. Although compensation is supported when clocked from the LPO, TCR[4:0] of the compensation register are also ignored and only TCR[7:5] set the compensation value (can overflow after 1020 to 1027 cycles). 0b - RTC prescaler increments using 32.768 kHz clock. 1b - RTC prescaler increments using 16.384 kHz LPO. Bit [0] of the prescaler is ignored.
6 —	Reserved
5 —	Reserved
4 —	Reserved
3 UM	Update Mode Allows SR[TCE] to be written even when the Status Register is locked. When set, the SR[TCE] can be written if the SR[TIF] or SR[TOF] are set or if the SR[TCE] is clear. Whenever both SR[TCE] and CR[UM] are set, then SR[TCE] should only be written once either SR[TIF] or SR[TOF] are set. 0b - Registers cannot be written when locked. 1b - Registers can be written when locked under limited conditions.
2 —	Reserved
1 —	Reserved

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
0 SWR	Software Reset 0b - No effect. 1b - Resets all RTC registers except for this bit . This bit is cleared by POR and by software explicitly clearing it.

27.5.1.7 RTC Status (SR)

Offset

Register	Offset
SR	14h

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								0	0		TCE	0	TAF	TOF	TIF
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Fields

Field	Function
31-8 —	Reserved
7 —	Reserved
6-5 —	Reserved
4 TCE	Time Counter Enable Specifies whether to enable or disable time counter:

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	0b - Disables. When time counter is disabled the TSR register and TPR register are writeable, but do not increment. 1b - Enables. When time counter is enabled the TSR register and TPR register are not writeable, but increment.
3 —	Reserved
2 TAF	Time Alarm Flag • Is set when the TAR[TAR] equals the TSR[TSR] and the TSR[TSR] increments. • Is cleared by writing the TAR register. • Indicates whether time alarm has occured: 0b - Time alarm has not occurred. 1b - Time alarm has occurred.
1 TOF	Time Overflow Flag • Is set when the time counter is enabled and overflows. The TSR and TPR do not increment and read as zero when this bit is set. • Is cleared by writing the TSR register when the time counter is disabled. • Indicates whether time overflow has occured: 0b - Time overflow has not occurred. 1b - Time overflow has occurred and time counter reads as zero.
0 TIF	Time Invalid Flag • The time invalid flag is set on POR or software reset. • The TSR and TPR do not increment and read as zero when this bit is set. • This bit is cleared by writing the TSR register when the time counter is disabled. • Indicates whether the time is valid or invalid: 0b - Time is valid. 1b - Time is invalid and time counter is read as zero.

27.5.1.8 RTC Lock (LR)

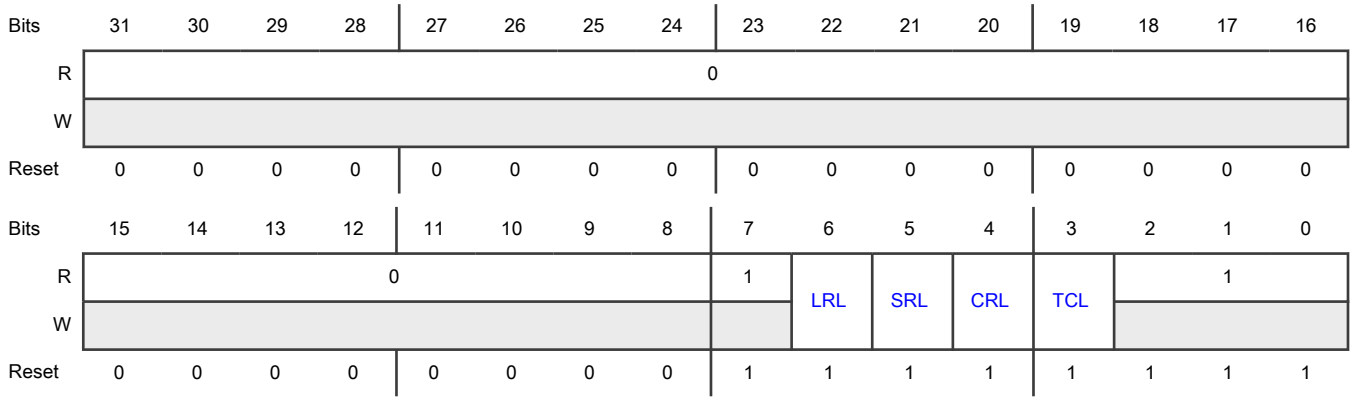
Offset

Register	Offset
LR	18h

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram



Fields

Field	Function
31-8 —	Reserved
7 —	Reserved
6 LRL	Lock Register Lock After being cleared, this bit can be set only by POR or software reset. 0b - Lock Register is locked and writes are ignored. 1b - Lock Register is not locked and writes complete as normal.
5 SRL	Status Register Lock After being cleared, this bit can be set only by POR or software reset. 0b - Status Register is locked and writes are ignored. 1b - Status Register is not locked and writes complete as normal.
4 CRL	Control Register Lock After being cleared, this bit can only be set by POR. 0b - Control Register is locked and writes are ignored. 1b - Control Register is not locked and writes complete as normal.
3 TCL	Time Compensation Lock After being cleared, this bit can be set only by POR or software reset. 0b - Time Compensation Register is locked and writes are ignored. 1b - Time Compensation Register is not locked and writes complete as normal.
2-0 —	Reserved

All information provided in this document is subject to legal disclaimers.

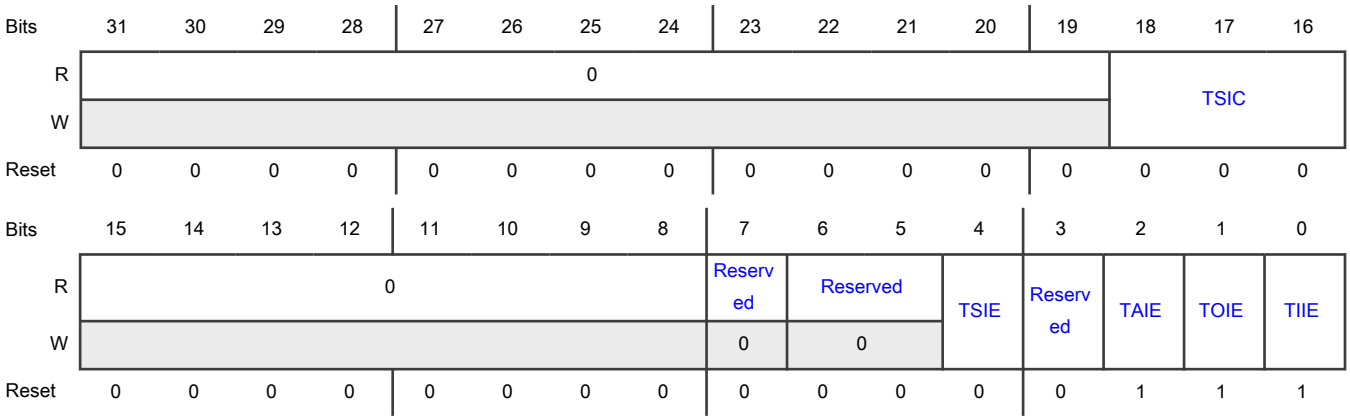
© 2026 NXP B.V. All rights reserved.

27.5.1.9 RTC Interrupt Enable (IER)

Offset

Register	Offset
IER	1Ch

Diagram



Fields

Field	Function
31-19 —	Reserved
18-16 TSIC	Timer Seconds Interrupt Configuration Configures the frequency of the RTC Seconds interrupt and the RTC_CLKOUT prescaler output. This field should only be altered when TSIE is clear. 000b - 1 Hz. 001b - 2 Hz. 010b - 4 Hz. 011b - 8 Hz. 100b - 16 Hz. 101b - 32 Hz. 110b - 64 Hz. 111b - 128 Hz.
15-8 —	Reserved
7	Reserved

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
—	
6-5 —	Reserved
4 TSIE	Time Seconds Interrupt Enable • Is an edge-sensitive interrupt with a dedicated interrupt vector. It is generated at least once a second and requires no software overhead (there is no corresponding status flag to clear). • The frequency of the seconds interrupt is configured by TSIC. • Specifies whether to enable or disable the time seconds interrupt: 0b - Disables 1b - Enables
3 —	Reserved
2 TAIE	Time Alarm Interrupt Enable Specifies whether an interrupt is generated by the time alarm flag: 0b - No interrupt is generated. 1b - An interrupt is generated.
1 TOIE	Time Overflow Interrupt Enable Specifies whether an interrupt is generated by the time overflow flag: 0b - No interrupt is generated. 1b - An interrupt is generated.
0 TIIE	Time Invalid Interrupt Enable Specifies whether an interrupt is generated by the time invalid flag: 0b - No interrupt is generated. 1b - An interrupt is generated.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Chapter 28
Enhanced Flex Pulse Width Modulator (eFlexPWM)

28.1 Chip-specific PWM information

Table 158. Reference links to related information

Topic	Related module	Reference
Full description	PWM	PWM
System memory map		Memory map
Clocking		Clock distribution
Power Management		Power Management
Bus and Memories		Memory architecture
Signal multiplexing	Port control	Signal multiplexing

28.1.1 Module instances

This chip supports one instance of the FlexPWM module: FlexPWM0. It has 3 sub-modules.

28.2 Overview

The Enhanced Flex Pulse Width Modulator (eFlexPWM) module contains PWM submodules, each of which can be used to control a single half-bridge power stage. Fault channel support is provided.

eFlexPWM generates various switching patterns, including highly sophisticated waveforms. It is ideal for controlling different Switched Mode Power Supplies (SMPS) topologies.

28.2.1 Block diagram

The following figure shows the eFlexPWM block diagram.

All information provided in this document is subject to legal disclaimers.
© 2026 NXP B.V. All rights reserved.

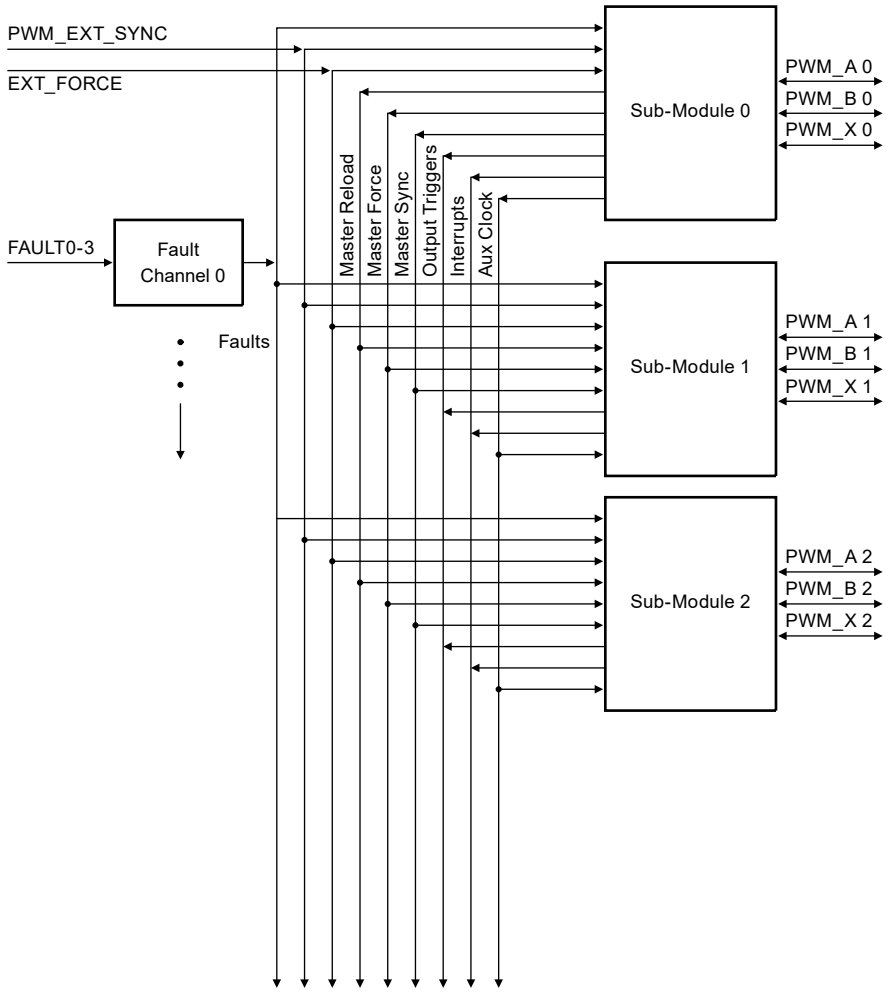


Figure 92. eFlexPWM block diagram

28.2.2 Features

Following are the features of eFlexPWM:

- 16-bit resolution for center, edge-aligned, and asymmetrical PWMs
- PWM outputs that can operate as complementary pairs or independent channels
- Ability to accept signed numbers for PWM generation
- Independent control of both edges of each PWM output
- Support for synchronization to external hardware or other PWM
- Double buffered PWM registers
  - Integral reload rates from 1 to 16
  - Half cycle reload capability
- Multiple output trigger events can be generated per PWM cycle via hardware
- Support for double switching PWM outputs
- Fault inputs can be assigned to control multiple PWM outputs
- Programmable filters for fault inputs

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

- Independently programmable PWM output polarity
- Independent top and bottom deadtime insertion
- Each complementary pair can operate with its own PWM frequency and deadtime values
- Individual software control for each PWM output
- All outputs can be programmed to change simultaneously via a FORCE_OUT event
- PWM_X pin can optionally output a third PWM signal from each submodule
- Channels not used for PWM generation can be used for buffered output compare functions and for input capture functions
- Enhanced dual edge capture functionality

28.3 Functional description

28.3.1 eFlexPWM submodule

The following figure shows the Submodule Block Diagram of eFlexPWM.

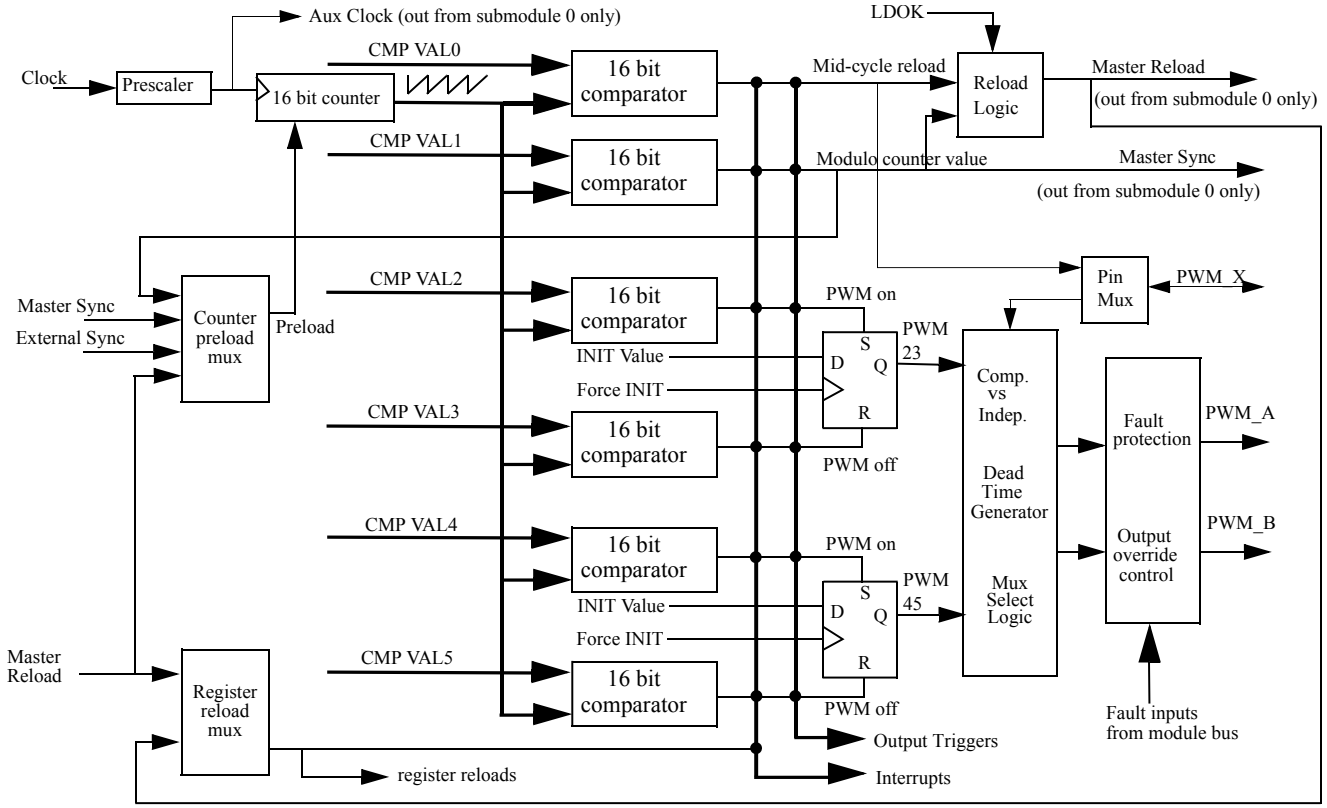


Figure 93. Submodule block diagram

28.3.2 eFlexPWM capabilities

This section describes some capabilities of the eFlexPWM module.

28.3.2.1 Center aligned PWMs

Each submodule has its own timer that is capable of generating PWM signals on two output pins. The edges of each of these signals are controlled independently as shown in Figure 94.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

NOTE

PWM's output period length is (VAL1 - INIT) PWM clock period plus 1 IPG clock period. PWM clock is the output of IPG clock after prescale processing.

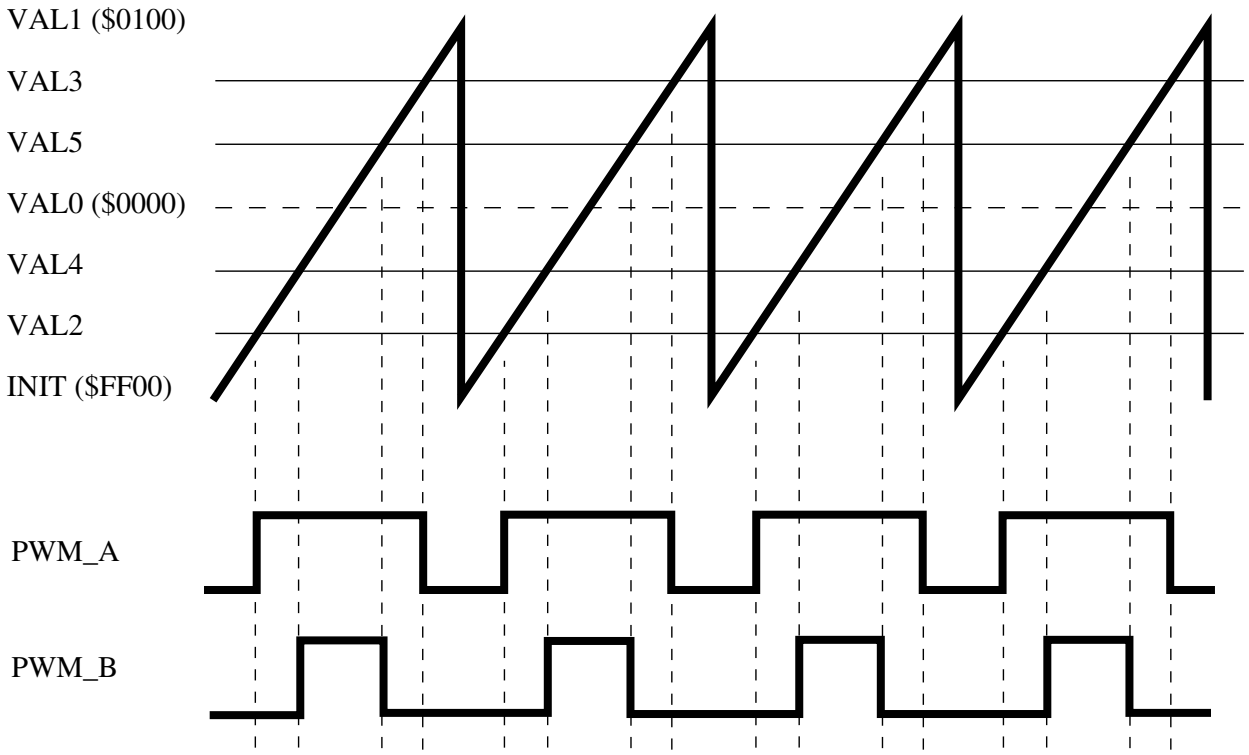


Figure 94. Center aligned PWM example

The submodule timers only count in the up direction and then reset to the INIT value. Instead of having a single value that determines pulse width, there are two values that must be specified: the turn-on edge and the turn-off edge. This double-action edge generation provides the user control over the pulse width and also the relative alignment of the signal. As a result, there is no need to support separate PWM alignment modes since the PWM alignment mode is inherently a function of the turn-on and turn-off edge values.

Figure 94 also illustrates an additional enhancement to the PWM generation process. When the counter resets, it is reloaded with a user-specified value, which may or may not be zero. If the value chosen happens to be the 2's complement of the modulus value, then the PWM generator operates in "signed" mode. This means if each PWM's turn-on and turn-off edge values are same in numbers but different in their sign, the "on" portion of the output signal is centered around a count value of zero. Therefore, only one PWM value is calculated in software and then this value and its negative are provided to the submodule as the turn-off and turn-on edges respectively. This technique results in a pulse width consists of an odd number of timer counts. If all PWM signal edge calculations follow this convention, then the signals will be center aligned with each other, which is the goal. The center alignment between the signals is not restricted to symmetry around the zero count value, as any other number would also work. However, centering on zero provides the greatest range in signed mode and also simplifies the calculations.

28.3.2.2 Edge aligned PWMs

Figure 95 shows the results of edge aligned operation when the turn-on edge for each pulse is specified to be the INIT value. Therefore, only the turn-off edge value needs to be periodically updated to change the pulse width.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

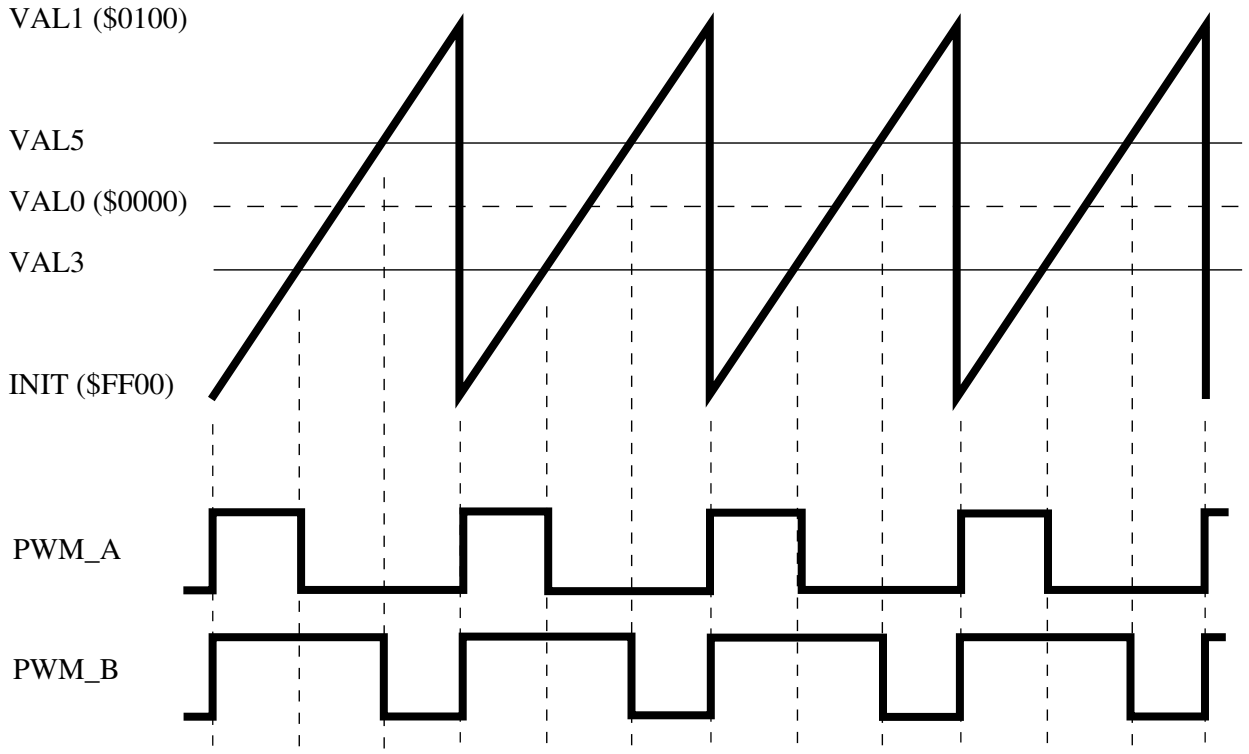


Figure 95. Edge aligned example (INIT=VAL2=VAL4)

With edge aligned PWMs, another example of the benefits of signed mode can be seen. Use "bipolar" PWMs to drive an H-bridge, where a 50% duty cycle results in zero volts on the load. Duty cycles less than 50% generate negative load voltages and duty cycles greater than 50% generate positive load voltages. If the module is set to signed mode operation (the INIT and VAL1 values are the same number with opposite signs), then there is a direct proportionality between the PWM turn-off edge value and the motor inverter voltage, including the sign. Therefore, signed mode of operation simplifies the software interface to the eFlexPWM module since no offset calculations are required to translate the output variable control algorithm to the voltage on an H-Bridge load.

28.3.2.3 Phase shifted PWMs

In the previous sections, the benefits of the signed mode of operation were discussed in the context of simplifying the required software calculations by eliminating the requirement to bias up signed variables before applying them to the module. However, if numerical biases are applied to the turn-on and turn-off edges of different PWM signals, the signals will be phase shifted to each other, as shown in Figure 96. This results in certain advantages when applied to a power stage. For example, when operating a multi-phase inverter at a low modulation index, all of the PWM switching edges from the different phases occur at the same time. This can be troublesome from a noise standpoint, especially if ADC readings of the inverter must be scheduled near those times. Phase shifting the PWM signals can open up timing windows between the switching edges to allow a signal to be sampled by the ADC. However, phase shifting does not affect the duty cycle so average load voltage is not affected.

If the outputs of submodules 1 - 2 need to be delayed from the output of submodule 0 (and from each other), instead of just creating a phase delay by adding an offset to the turn on and turn off times of the different submodules, another method is to use the PHASEDLY registers for submodules 1 - 2 to indicate their delay from the submodule 0 timing. This method can be used when the master sync signal from submodule 0 is selected as the initialization source (CTRL2[INIT_SEL]==b10). This method allows all of the submodules to be programmed with the same turn on and turn off time but submodules 1 - 2 can still be delayed the time from submodule 0.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

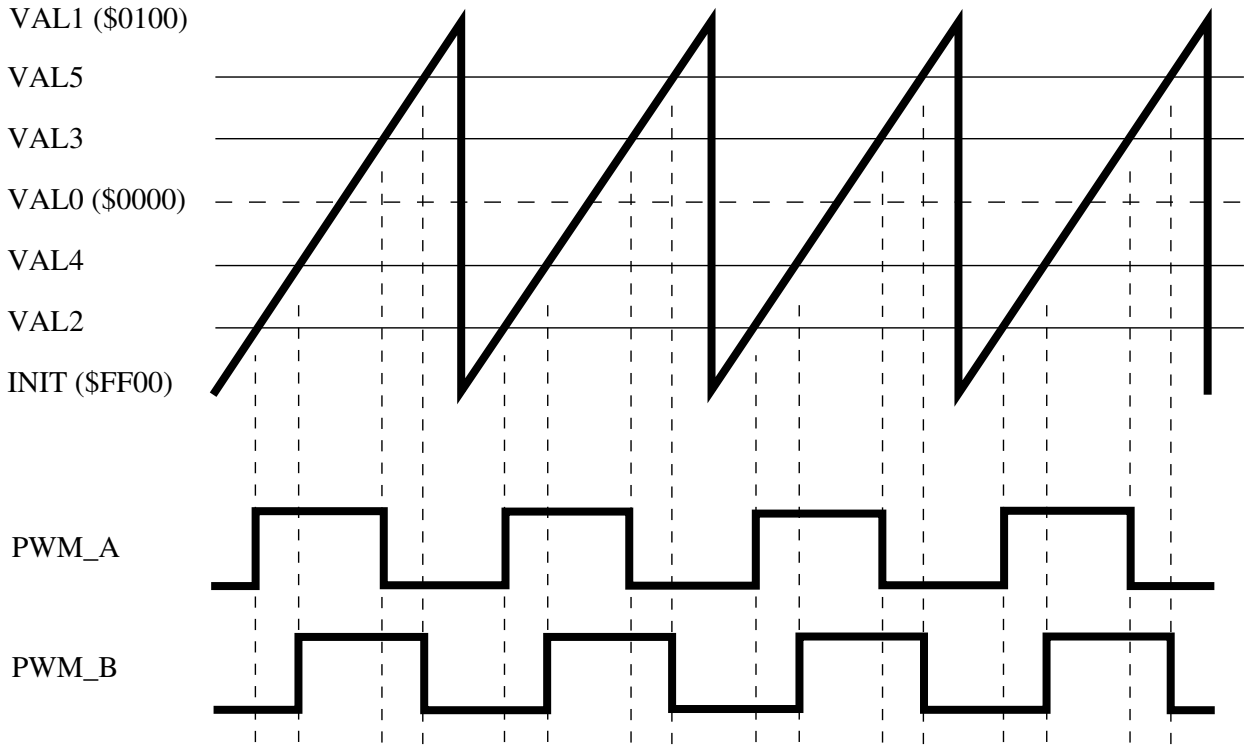


Figure 96. Phase shifted outputs - example for 1 submodule

An additional benefit of phase shifted PWMs is shown in Figure 97. In this case, an H-Bridge circuit is driven by 4 PWM signals to control the voltage waveform on the primary of a transformer. Both left and right side PWMs are configured to generate a square wave with 50% duty cycle. This works for the H-Bridge since no narrow pulse widths are generated reducing the high frequency switching requirements of the transistors. The RMS value of this waveform is directly controlled by the amount of phase shift of the square waves. Regardless of the phase shift, no DC component appears in the load voltage as long as the duty cycle of each square wave remains at 50% suitable for transformer loads. As a result, this topology is frequently used in industrial welders to adjust the amount of energy delivered to the weld arc.

NOTE

The square wave on the right side of the H-Bridge is phase shifted compared to the left side of the H-Bridge. As a result, the transformer primary sees the bottom waveform across its terminals.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

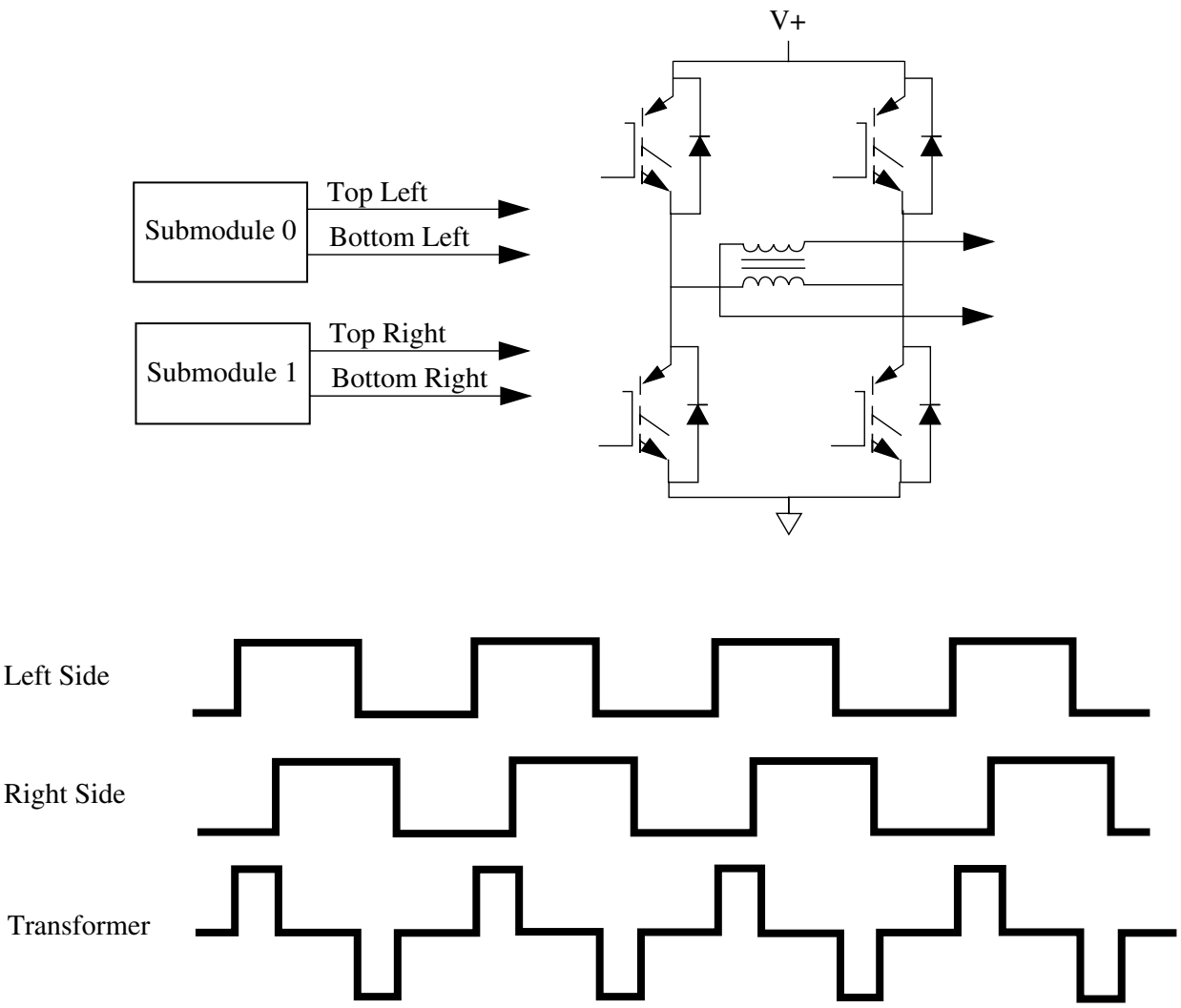


Figure 97. Phase Shifted PWMs Applied to a Transformer Primary

28.3.2.4 Double switching PWMs

Double switching PWM output is supported to aid in single shunt current measurement and three-phase reconstruction. This method supports two independent rising edges and two independent falling edges per PWM cycle. The VAL2 and VAL3 registers are used to generate the even channel (labeled as PWM_A in Figure 98) while VAL4 and VAL5 are used to generate the odd channel. The two channels (PWM23 or PWM_A and PWM45 or PWM_B from force out logic) are combined using XOR logic (force out logic) as the following figure shows. The DBLPWM signal can be run through the deadtime insertion logic.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

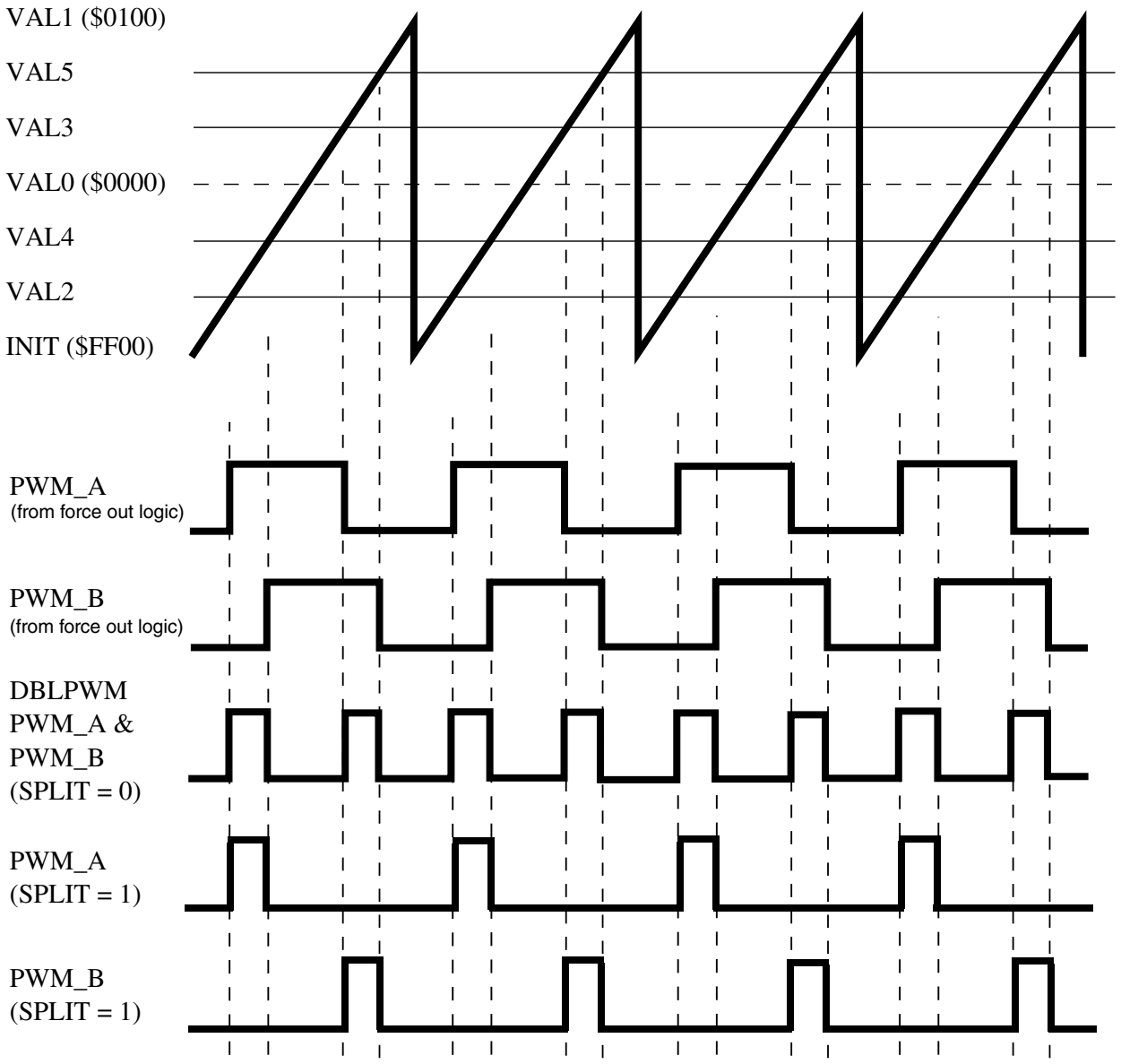


Figure 98. Double switching output example

28.3.2.5 ADC triggering

In cases where the timing of the ADC triggering is critical, it must be scheduled as a hardware event instead of software activated. With this eFlexPWM module, multiple ADC triggers can be generated in hardware per PWM cycle without the requirement of another timer module. Figure 99 shows how this is accomplished. When specifying a complementary mode of operation, only two edge comparators are required to generate the output PWM signals for a given submodule. This means the other comparators are free to perform other functions. In this example, the software does not need to quickly respond after the first conversion to set up other conversions that must occur in the same PWM cycle.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

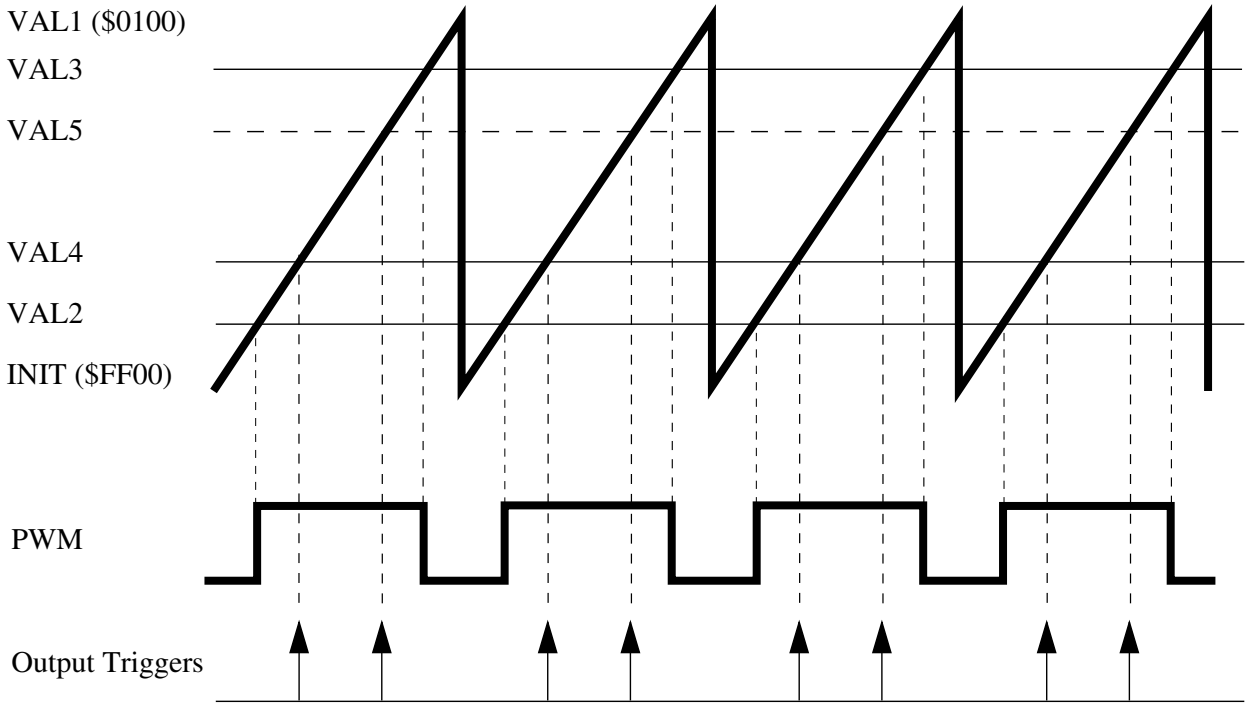


Figure 99. Multiple output trigger generation in hardware

Because each submodule has its own timer, it is possible for each submodule to run at a different frequency. One of the options possible with this eFlexPWM module is to have one or more submodules running at a lower frequency, but still synchronized to the timer in submodule0. Figure 100 shows how this feature can be used to schedule ADC triggers over multiple PWM cycles. You can use the lower-frequency submodule to control the sampling frequency of the software control algorithm where multiple ADC triggers can now be scheduled over the entire sampling period. In Figure 100, *all* submodule comparators are shown being used for ADC trigger generation.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

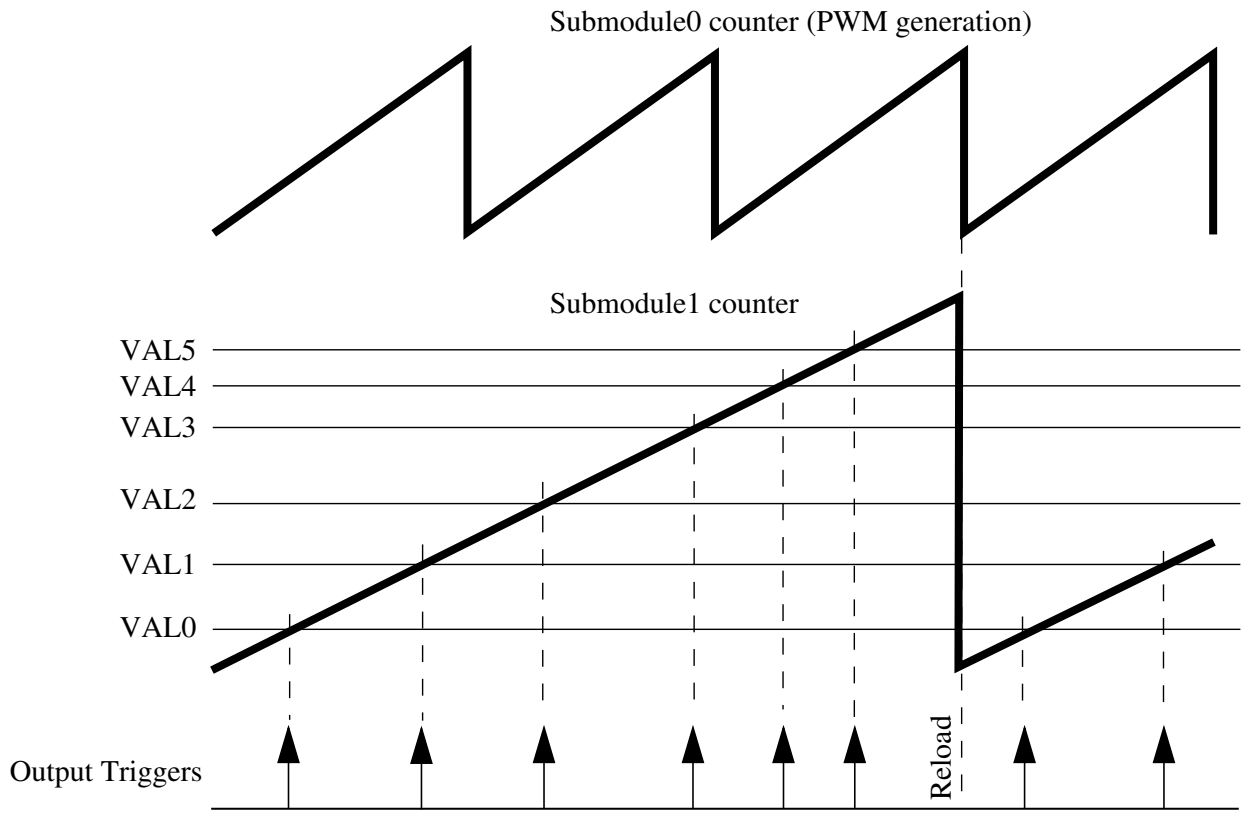


Figure 100. Multiple output triggers over several PWM cycles

28.3.2.6 Enhanced capture capabilities (E-Capture)

When a PWM pin is not used for PWM generation, it can be used to perform input captures. For PWM generation, both edges of the PWM signals are specified via separate compare register values. When programmed for input capture, both of these registers work on the same pin to capture multiple edges, toggling from one to the other in either a free running or one-shot fashion. Programming the desired edge of each capture circuit, period, and pulse width of an input signal can easily be measured without the requirement to re-arm the circuit. In addition, each edge of the input signal can clock an 8-bit counter where the counter output is compared to a user specified value (EDGCMP). When the counter output equals EDGCMP, the value of the submodule timer is captured and the counter is automatically reset. This feature counts a specified number of edge events and then performs a capture and interrupt. The following figure illustrates some of the functionality of the E-Capture circuit.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

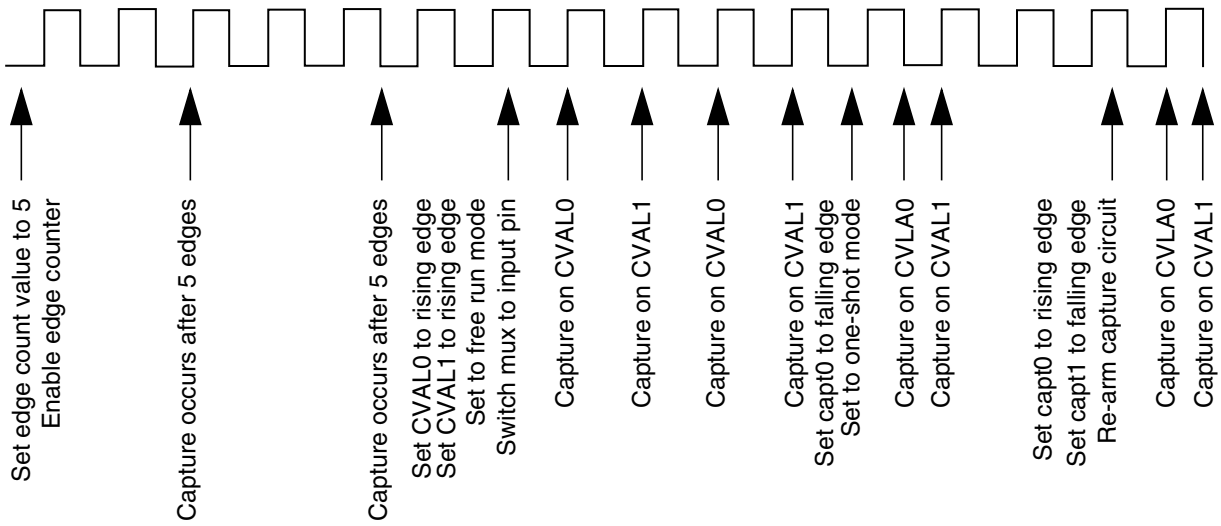


Figure 101. Capture capabilities of the E-Capture circuit

When a submodule is used for PWM generation, its timer counts up to the modulus value used to specify the PWM frequency and then is re-initialized. Therefore, using this timer for input captures on one of the other pins (for example, PWM_X) has limited utility since it does not count through all of the numbers and the timer reset represents a discontinuity in the 16-bit number range. However, when measuring a signal that is synchronous to the PWM frequency, the timer modulus range is suited for the application. As shown in Figure 102, the output of a PWM power stage is connected to the PWM_X pin that is configured for free running input captures. Specifically, the CVAL0 capture circuitry is programmed for rising edges and the CVAL1 capture circuitry is set for falling edges. This results in new load pulse width data acquired every PWM cycle. To calculate the pulse width, subtract the CVAL0 register value from the CVAL1 register value. This measurement is beneficial when performing deadtime distortion correction on a half bridge circuit driving an inductive load. Also, these values can be directly compared to the VALx registers responsible for generating the PWM outputs to obtain a measurement of system propagation delays. For details, refer to the separate discussion of deadtime distortion correction.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

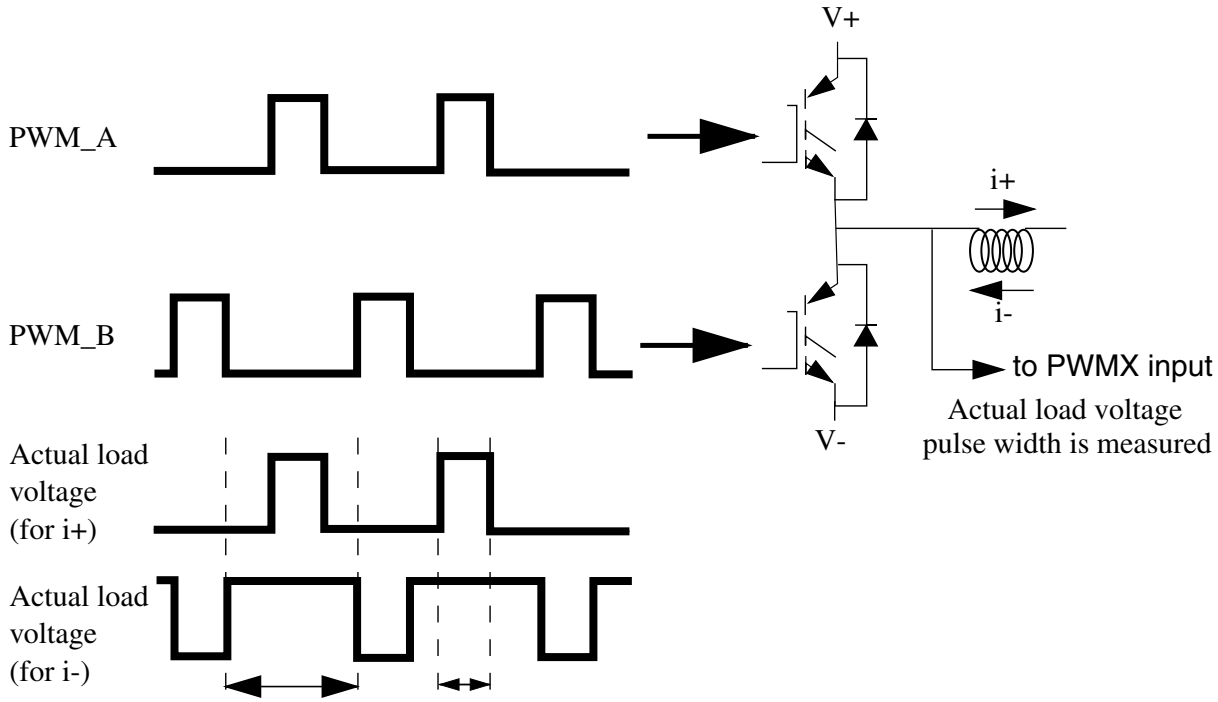


Figure 102. Output pulse width measurement possible with the E-capture circuit

28.3.2.7 Synchronous switching of multiple outputs

Before the PWM signals are routed to the output pins, they are processed by a hardware block that permits all submodule outputs to be switched synchronously. This feature is useful in commutated motor applications where the next commutation state can be laid in ahead of time and then immediately switched to the outputs when the appropriate condition or time is reached. All the changes occur immediately after the trigger event occurs eliminating any interrupt latency and also the changes occurs synchronously on all submodule outputs.

The synchronous output switching is accomplished via a signal called FORCE_OUT. This signal originates from the local FORCE bit within the submodule, from submodule0, or from external to the eFlexPWM module, and in most cases, is supplied from an external timer channel configured for output compare. In a typical application, software sets up the desired states of the output pins in preparation for the next FORCE_OUT event. This selection lays dormant until the FORCE_OUT signal transitions and then all outputs are switched simultaneously. The signal switching is performed upstream from the deadtime generator so that any abrupt changes that occur do not violate deadtime on the power stage when in complementary mode.

Figure 103 shows an application that can benefit from this feature. On a brushless DC motor in many cases, it is required to spin the motor without need of hall-effect sensor feedback. Instead, the back EMF of the motor phases is monitored and this information is used to schedule the next commutation event. The top waveforms of Figure 103 represent these back EMF signals. Timer compare events (represented by the long vertical lines in the diagram) are scheduled based on the zero crossings of the back-EMF waveforms. The eFlexPWM module is configured via software ahead of time with the next state of the PWM pins in anticipation of the compare event. When it happens, the output compare of the timer drives the FORCE_OUT signal which immediately changes the state of the PWM pins to the next commutation state with no software latency.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

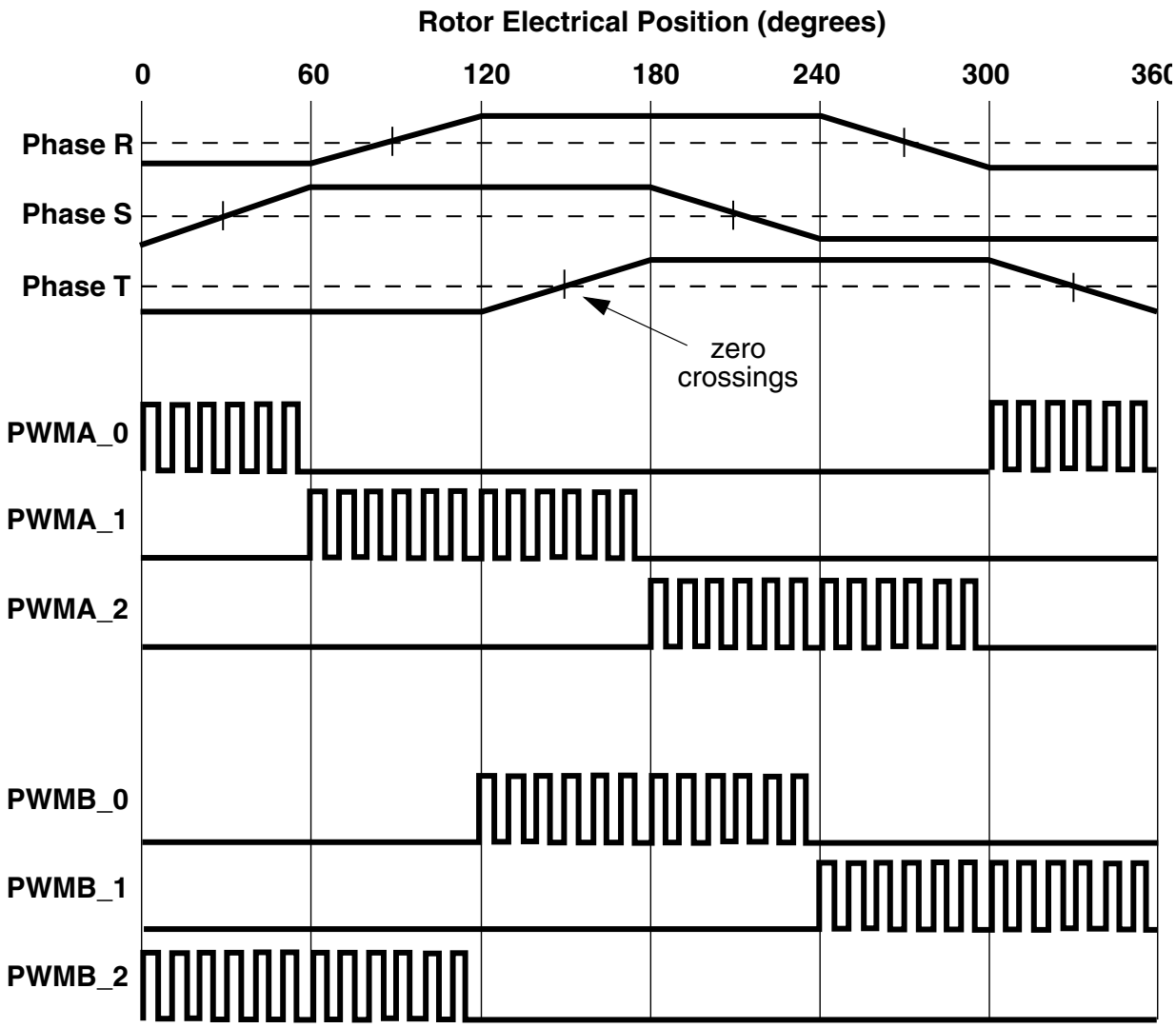


Figure 103. Sensorless BLDC commutation using the force out function

28.3.3 Operation

This section describes the implementation of various sections of eFlexPWM in detail.

The following figure is a high-level block diagram of output PWM generation.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

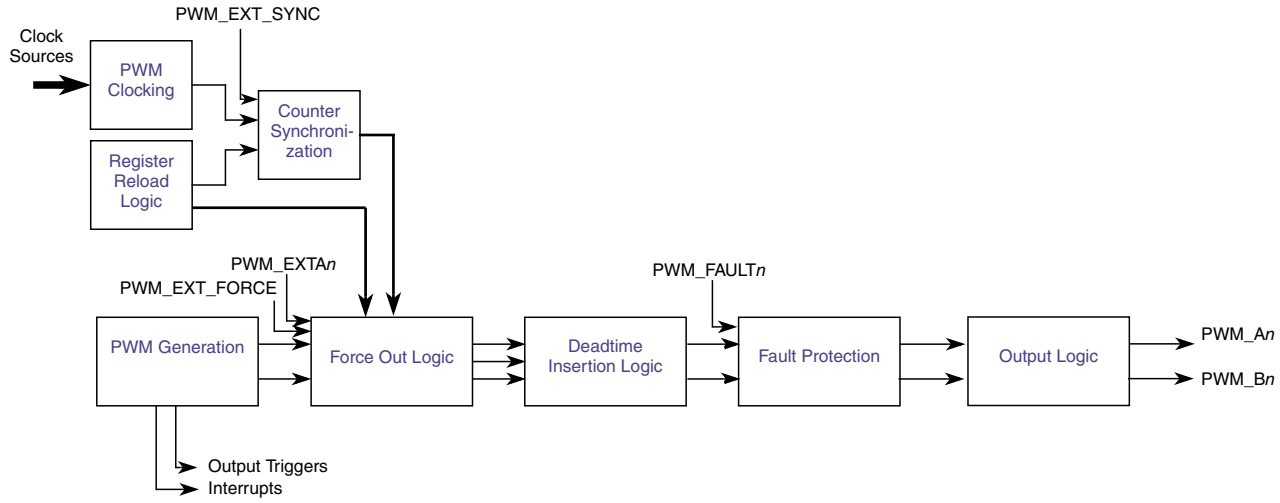


Figure 104. High-level output PWM generation block diagram

28.3.3.1 Register reload logic

The register reload logic is used to determine when the outer set of registers for all double buffered register pairs will be transferred to the inner set of registers. The register reload event can be scheduled to occur every "n" PWM cycles using CTRL[LDFQ] and CTRL[FULL], which is defined by VAL1 register. A half cycle reload option is also supported (CTRL[HALF]) where the reload can take place in the middle of a PWM cycle. The half cycle point is defined by the VAL0 register and does not have to be exactly in the middle of the PWM cycle.

As shown in Figure 105 the reload signal from submodule0 can be broadcast as the Master Reload signal allowing the reload logic from submodule0 to control the reload of registers in other submodules.

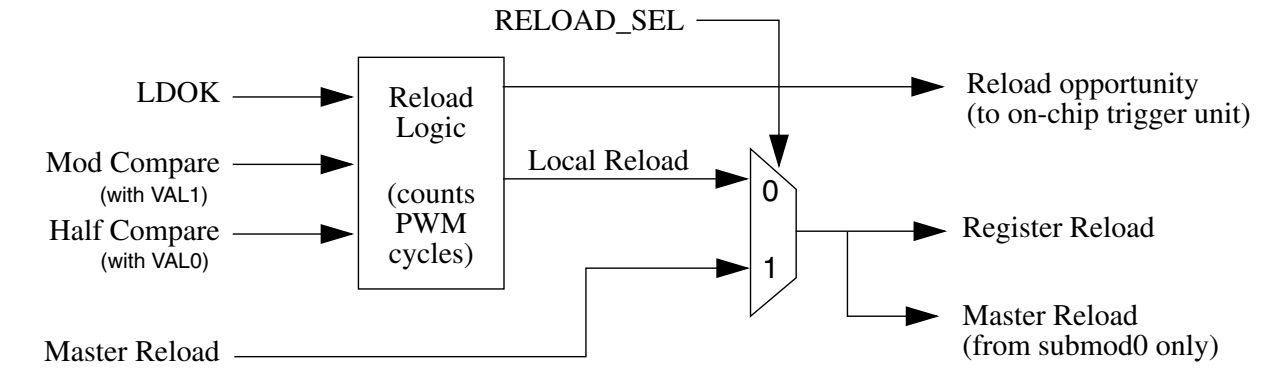


Figure 105. Register reload logic of each submodule

28.3.3.2 Counter synchronization

As shown in Figure 106, the 16-bit counter counts up until its output equals VAL1 which is used to specify the counter modulus value. The resulting compare causes a rising edge to occur on the Local Synchronization signal which is one of four possible sources used to cause the 16-bit counter to be initialized with INIT. If Local Synchronization is selected as the counter initialization signal, VAL1 within the submodule effectively controls the timer period (and then the PWM frequency generated by that submodule), and everything operates or functions at a local level.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

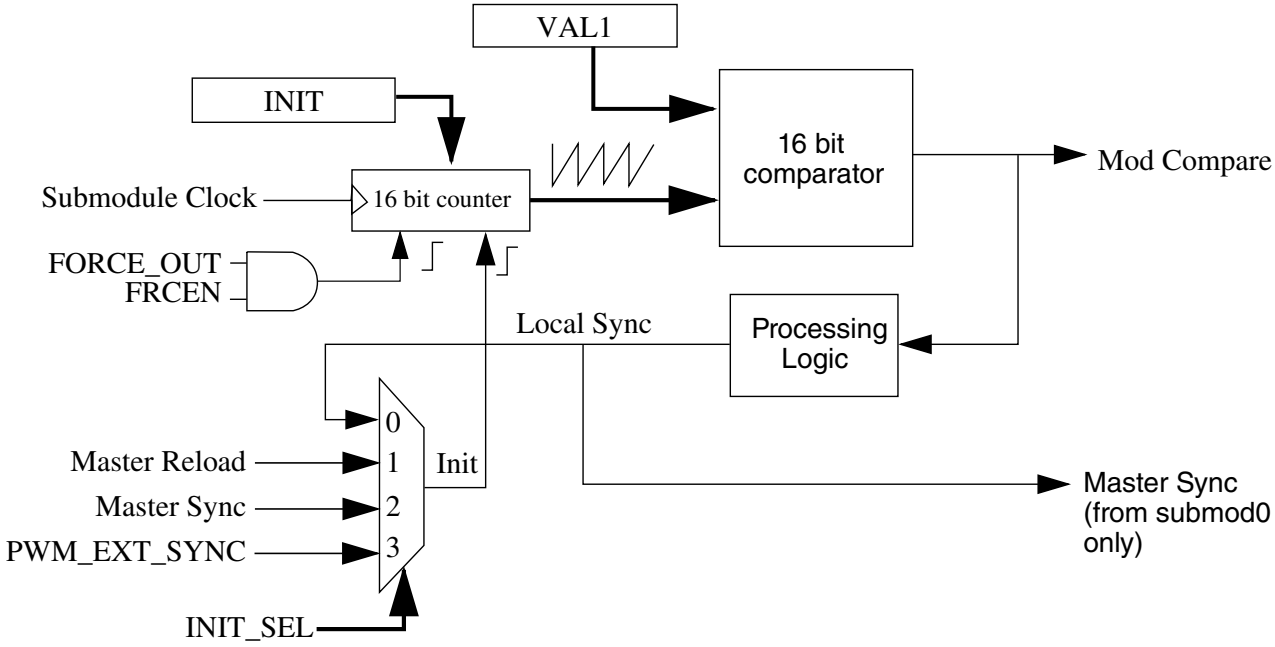


Figure 106. Counter synchronization for each submodule

The Master Synchronization signal originates as the Local Sync from submodule0. If configured to do so, the timer period of any submodule can be locked to the period of the timer in submodule0.

The PWM_EXT_SYNC signal originates on chip or off chip depending on the system architecture. This signal may be selected as the source for counter initialization so that an external source can control the period of all submodules.

If the Master Reload signal is selected as the source for counter initialization, then the period of the counter will be locked to the register reload frequency of submodule0. Since the reload frequency is commensurate to the sampling frequency of the software control algorithm, the submodule counter period is equal the sampling period. As a result, this timer can be used to generate output compares or output triggers over the entire sampling period which may consist of several PWM cycles. The Master Reload signal can only originate from submodule0.

The counter can optionally initialize upon the assertion of the FORCE_OUT signal assuming that CTRL2[FRCEN] is set. As shown in Figure 106, this constitutes a second initialization input into the counter, which causes the counter to initialize regardless of which signal is selected as the counter initialization signal. A forced initialization causes a register reload if MCTRL[LDOK] is set.

The counter can be initialized by FORCE_OUT signal only when MCTRL[RUN] = 1 or CTLR2[CLK_SEL] = 2 which chooses auxiliary clock from SM0.

The FORCE_OUT signal is provided mainly for commutated applications. When PWM signals are commutated on an inverter controlling a brushless DC motor, it is necessary to restart the PWM cycle at the beginning of the commutation interval. This action effectively resynchronizes the PWM waveform to the commutation timing. Otherwise, the average voltage applied to a motor winding integrated over the entire commutation interval will be a function of the timing between the asynchronous commutation event for the PWM cycle. The effect is more critical at higher motor speeds where each commutation interval may consist of only a few PWM cycles. If the counter is not initialized at the start of each commutation interval, the result is an oscillation caused by the beating between the PWM frequency and the commutation frequency.

28.3.3.3 PWM generation

Figure 107 illustrates how PWM generation is accomplished in each submodule. In each case, two comparators and associated VALx registers are utilized for each PWM output signal. One comparator and VALx register are used to control the turn-on edge, while a second comparator and VALx register control the turn-off edge.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

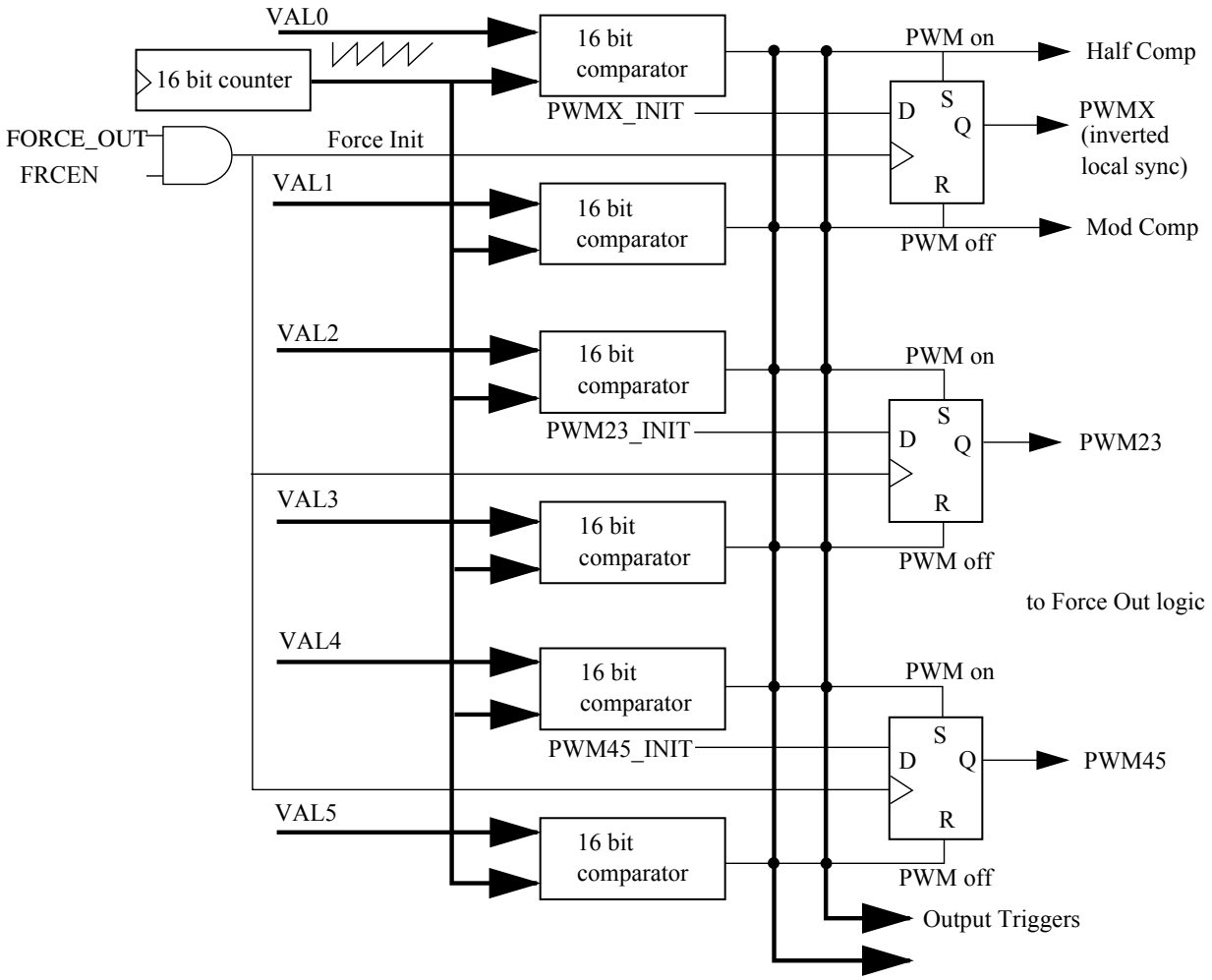


Figure 107. PWM Generation Hardware of each submodule

The generation of the Local Synchronization signal is performed exactly the same way as the other PWM signals in the submodule. While comparator 0 causes a falling edge of the Local Synchronization signal, comparator 1 generates a rising edge. Comparator 1 is also hardwired to the reload logic to generate the full cycle reload indicator.

If VAL1 is controlling the modulus of the counter and VAL0 is half of the VAL1 register minus the INIT value, then the half cycle reload pulse occurs exactly half way through the timer count period and the Local Synchronization will have a 50% duty cycle. On the other hand, if the VAL1 and VAL0 registers are not required for register reloading or counter initialization, they can be used to modulate the duty cycle of the Local Synchronization signal, effectively turning it into an auxiliary PWM signal (PWM_X) assuming that the PWM_X pin is not being used for another function such as input capture or deadtime distortion correction. Including the Local Synchronization signal, each submodule is capable of generating three PWM signals where software has complete control over each edge of each of the signals.

If the comparators and edge value registers are not required for PWM generation, they can be used for other functions such as output compares, generating output triggers, or generating interrupts at timed intervals.

The 16-bit comparators shown in Figure 107 are "equal to" comparators. In addition, if both the set and reset of the flip-flop are asserted, then the flop output goes to 0.

28.3.3.4 Output compare capabilities

By using the VALx registers in conjunction with the submodule timer and 16-bit comparators, buffered output compare functionality can be achieved with no additional hardware required. Specifically, the following output compare functions are possible:

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

- An output compare sets the output high
- An output compare sets the output low
- An output compare generates an interrupt
- An output compare generates an output trigger

In PWM generation, an output compare is initiated by programming a VALx register for a timer compare, which in turn causes the output of the D flip-flop to either set or reset. For example, if an output compare is desired on the PWM_A signal that sets it high, VAL2 would be programmed with the counter value where the output compare should take place. However, to prevent the D flip-flop from being reset again after the compare has occurred, the VAL3 register must be programmed to a value out of the modulus range of the counter. Therefore, a comparison that would result in resetting the D flip-flop output would never occur. Conversely, if an output compare is desired on the PWM_A signal that sets it low, the VAL3 register is programmed with the appropriate count value and the VAL2 register is programmed with a value out of the counter modulus range. Regardless of whether a high compare or low compare is programmed, an interrupt, or output trigger can be generated when the compare event occurs.

28.3.3.5 Force out logic

For each submodule, the software can select between eight signal sources for the FORCE_OUT signal depending on the chip architecture:

1. Local CTRL2[FORCE]
2. Master Force signal from submodule0
3. Local Reload signal
4. Master Reload signal from submodule0
5. Local Synchronization signal
6. Master Synchronization signal from submodule0
7. EXT_SYNC signal from on or off chip
8. EXT_FORCE signal from on or off chip

The local signals are used to change the signals on the output pins of the submodule without regard for synchronization with other submodules. However, if it is required that all signals on all submodule outputs change at the same time, the Master, EXT_SYNC, or EXT_FORCE signals must be selected.

Figure 108 illustrates the Force logic. The SEL23 and SEL45 fields each choose from one of four signals that can be supplied to the submodule outputs: the PWM signal, the inverted PWM signal, a binary level specified by software via the OUT23 and OUT45 bits, or the PWM_EXTA alternate external control signals. The selection can be determined ahead of time, and when a FORCE_OUT event occurs, these values are presented to the signal selection mux that immediately switches the requested signal to the output of the mux for further processing downstream.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

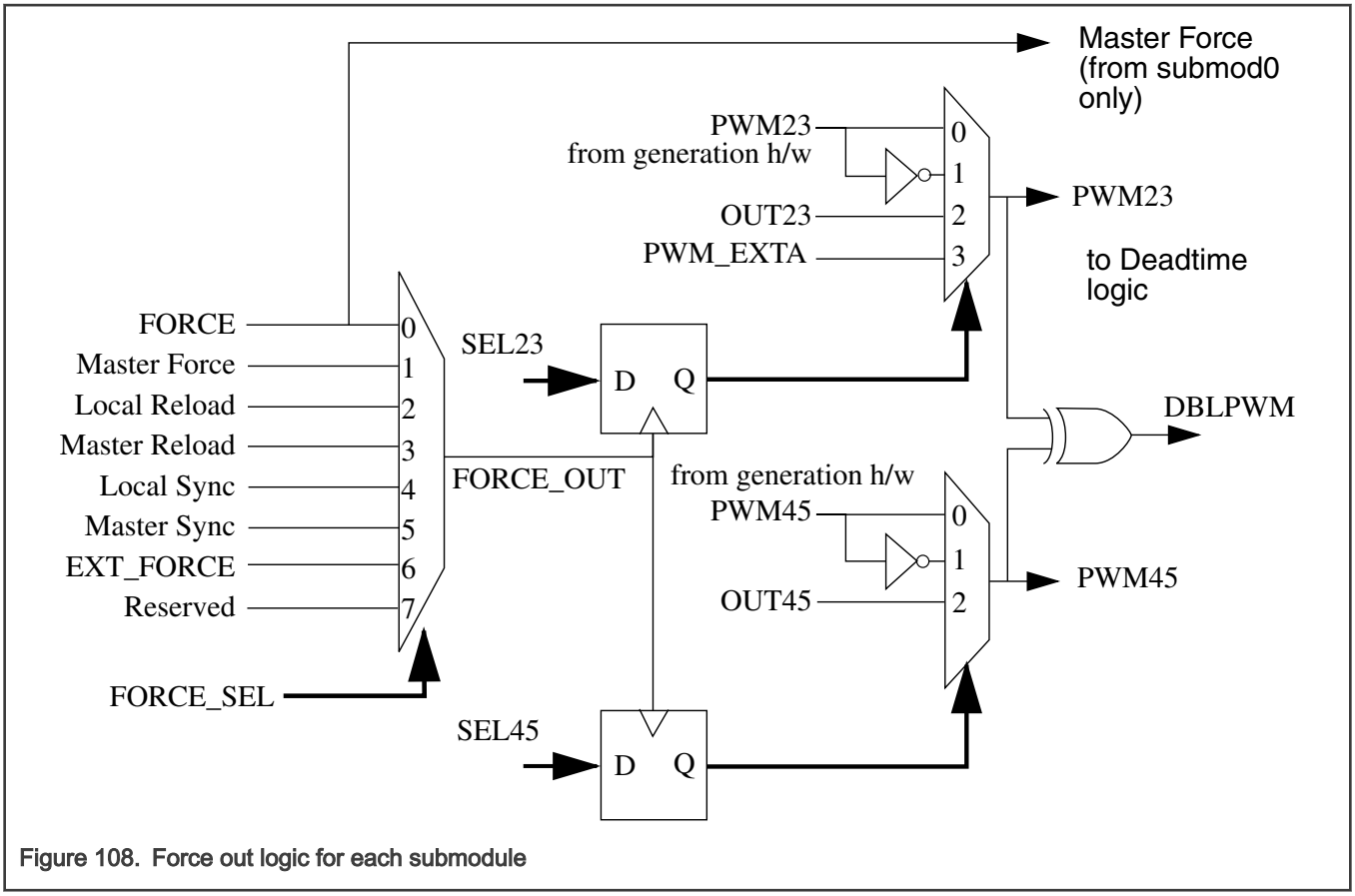


Figure 108. Force out logic for each submodule

The local CTRL2[FORCE] signal of submodule0 can be broadcast as the Master Force signal to other submodules. This feature allows the CTRL2[FORCE] of submodule0 to synchronously update all of the submodule outputs at the same time. The EXT_FORCE signal originates from outside the eFlexPWM module from a source such as a timer or digital comparators in the Analog-to-Digital Converter.

28.3.3.6 Independent or complementary channel operation

Writing a logic one to CTRL2[INDEP] configures the pair of PWM outputs as two independent PWM channels. Each PWM output is controlled by its own VALx pair operating independently of the other output.

Writing a logic zero to CTRL2[INDEP] configures the PWM output as a pair of complementary channels. The PWM pins are paired as shown in Figure 109 in complementary channel operation. The signal that is connected to the output pin (PWM23 or PWM45) is determined by MCTRL[IPOL].

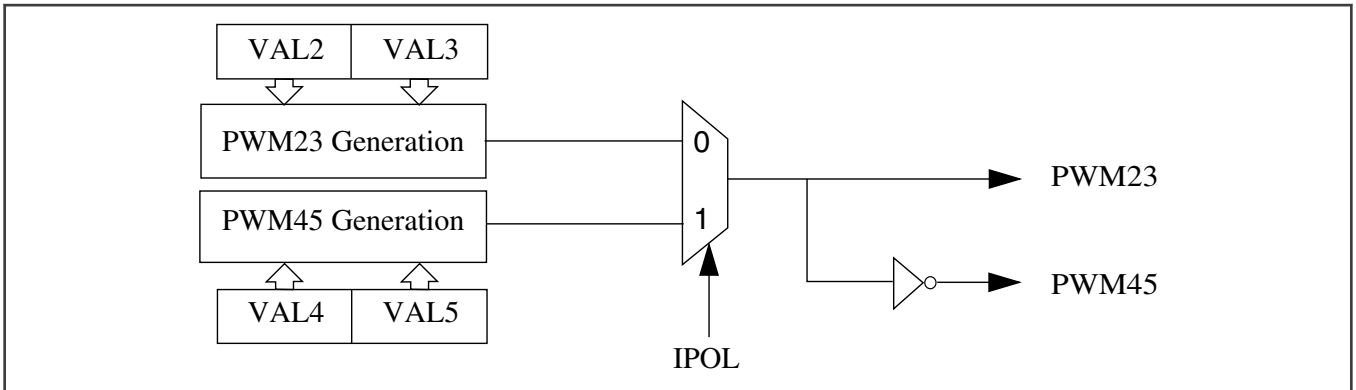


Figure 109. Complementary channel pair for each submodule

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

The complementary channel operation is for driving top and bottom transistors in a motor drive circuit, as shown in Figure 110.

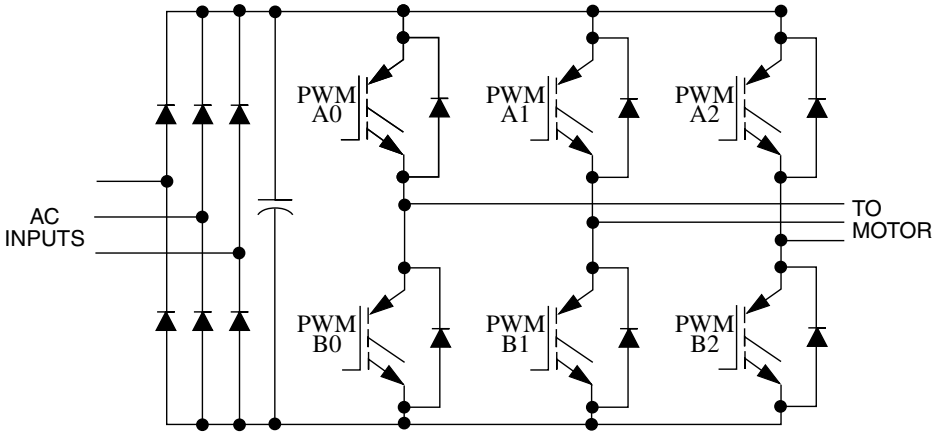


Figure 110. Typical 3 phase AC motor drive

The complementary operation allows the use of the deadtime insertion feature.

28.3.3.7 Deadtime insertion logic

The following figure shows the deadtime insertion logic of each submodule which is used to create non-overlapping complementary signals when not in independent mode.

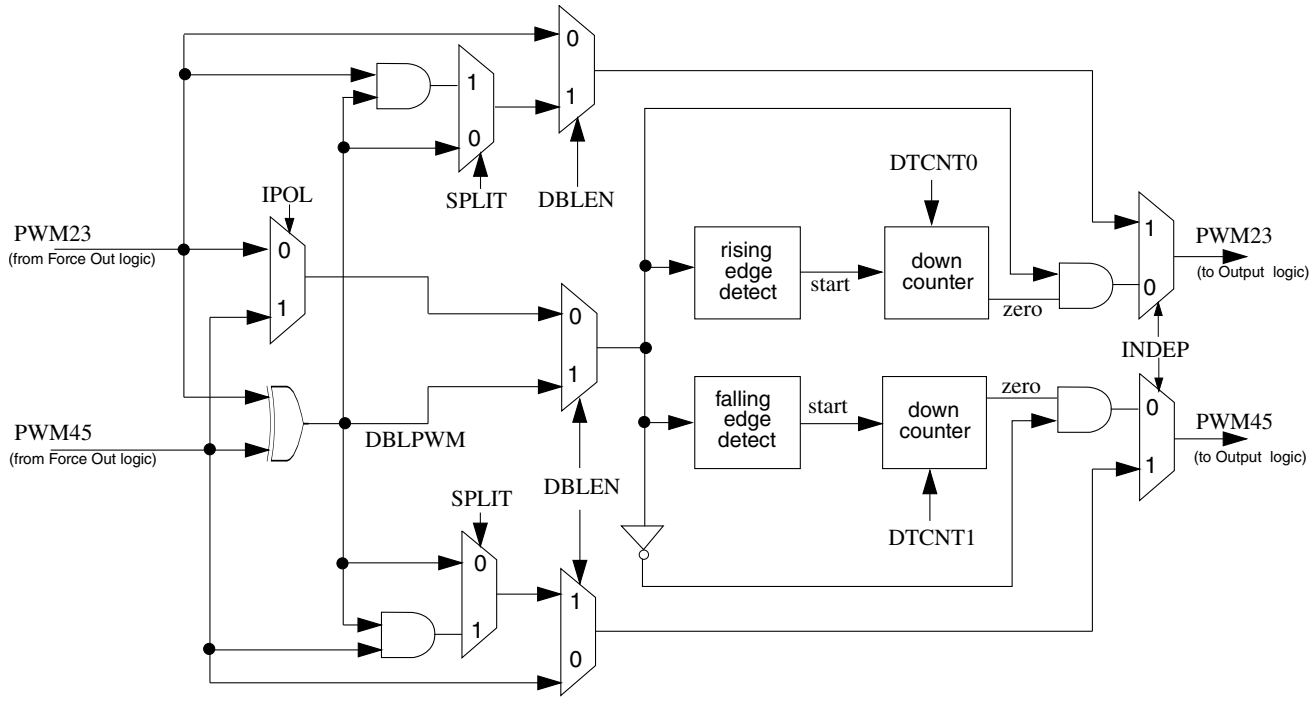


Figure 111. Deadtime Insertion Logic for each submodule

While in the complementary mode, a PWM pair can be used to drive top/bottom transistors, as shown in the figure. When the top PWM channel is active, the bottom PWM channel is inactive, and vice versa.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

NOTE

To avoid short-circuiting the DC bus and endangering the transistor, there must be no overlap of conducting intervals between the top and bottom transistors. But the transistor's characteristics make its switching-off time longer than switching-on time. To avoid the conducting overlap of top and bottom transistors, deadtime needs to be inserted in the switching period, as shown in Figure 112.

The deadtime generators automatically insert software-selectable activation delays into the pair of PWM outputs. The deadtime registers (DTCNT0 and DTCNT1) specify the number of IPBus clock cycles to use for deadtime delay. Every time the deadtime generator inputs change state, deadtime is inserted. Deadtime forces both PWM outputs in the pair to the inactive state.

When deadtime is inserted in complementary PWM signals connected to an inverter driving an inductive load, the PWM waveform on the inverter output will have a different duty cycle than what appears on the output pins of the eFlexPWM module. This results in a distortion in the voltage applied to the load. A method of correcting this, adding to or subtracting from the PWM value used, is discussed next.

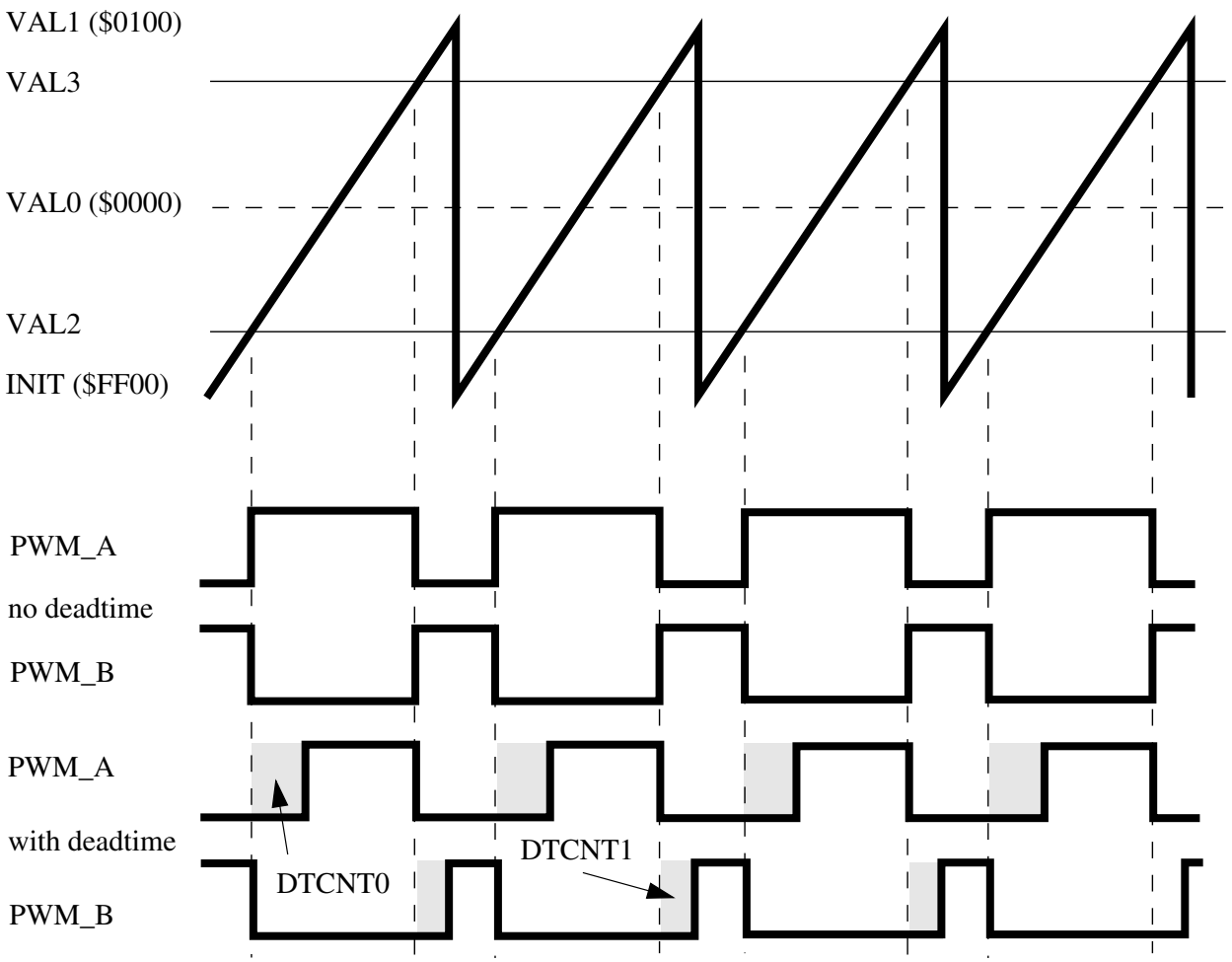


Figure 112. Deadtime insertion

28.3.3.7.1 Top/Bottom correction

In complementary mode, either the top or the bottom transistor controls the output voltage. However, deadtime has to be inserted to avoid overlap of conducting interval between the top and bottom transistor. Both transistors in complementary mode are off during deadtime, allowing the output voltage to be determined by the current status of load and introducing distortion in the output voltage, as shown in Figure 113. On AC induction motors running open-loop, the distortion typically manifests itself as poor low-speed performance, such as torque ripple and rough operation.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

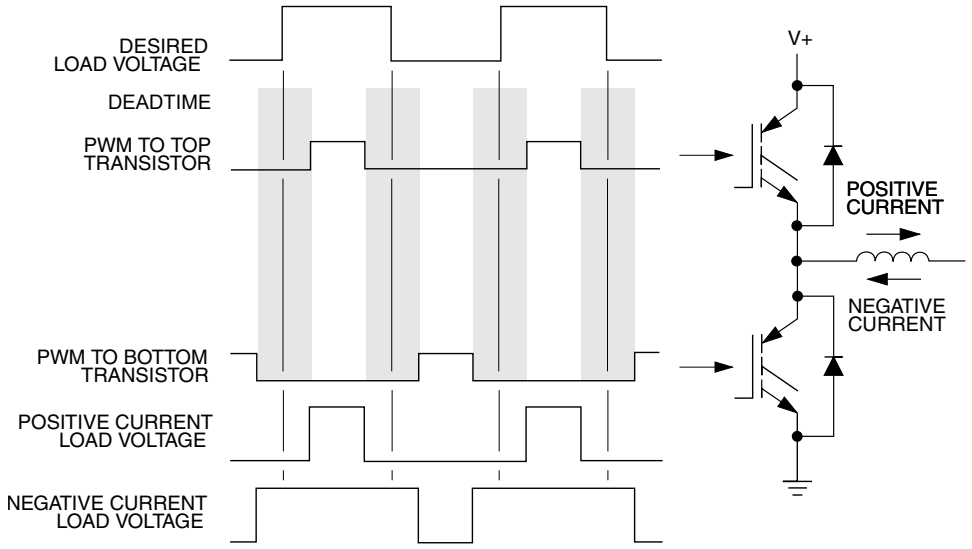


Figure 113. Deadtime distortion

During deadtime, load inductance distorts output voltage by keeping current flowing through the diodes. This deadtime current flow creates a load voltage that varies with the current direction. With a positive current flow, the load voltage during deadtime is equal to the bottom supply, putting the top transistor in control. With a negative current flow, the load voltage during deadtime is equal to the top supply putting the bottom transistor in control.

Remembering that the original PWM pulse widths were shortened by deadtime insertion, the averaged sinusoidal output is less than the desired value. However, when deadtime is inserted, it distorts in the motor current waveform inverter outputs. This distortion is aggravated by different turn-on and turn-off delays of each of the transistors. By giving the eFlexPWM module information on which transistor is controlling at a given time this distortion can be corrected.

For a typical circuit in complementary channel operation, only one of the transistors is effective in controlling the output voltage at any given time. This depends on the direction of the motor inverter current for that pair, as shown in Figure 113. To correct distortion one of two different factors must be added to the desired PWM value, depending on whether the top or bottom transistor is controlling the output voltage. Therefore, the software is responsible for calculating both compensated PWM values prior to placing them in the VALx registers. Either the VAL2/VAL3 or the VAL4/VAL5 register pair controls the pulse width at any given time. For a given PWM pair, whether the VAL2/VAL3 or VAL4/VAL5 pair is active depends on either:

- The state of the current status pin, PWMX, for that driver
- The state of the odd/even correction bit, MCTRL[IPOL], for that driver

To correct deadtime distortion, the software can decrease or increase the value in the appropriate VALx register.

- In edge-aligned operation, decreasing or increasing the PWM value by a correction value equal to the deadtime typically compensates for deadtime distortion.
- In center-aligned operation, decreasing or increasing the PWM value by a correction value equal to one-half the deadtime typically compensates for deadtime distortion.

28.3.3.7.2 Manual correction

To detect the current status, the voltage on each PWMX pin is sampled twice in a PWM period, at the end of each deadtime. The value is stored in CTRL[DT]. CTRL[DT] is a timing marker indicating when to toggle between PWM value registers. The software can then set MCTRL[IPOL] to switch between VAL2/VAL3 and VAL4/VAL5 register pairs according to CTRL[DT] values.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

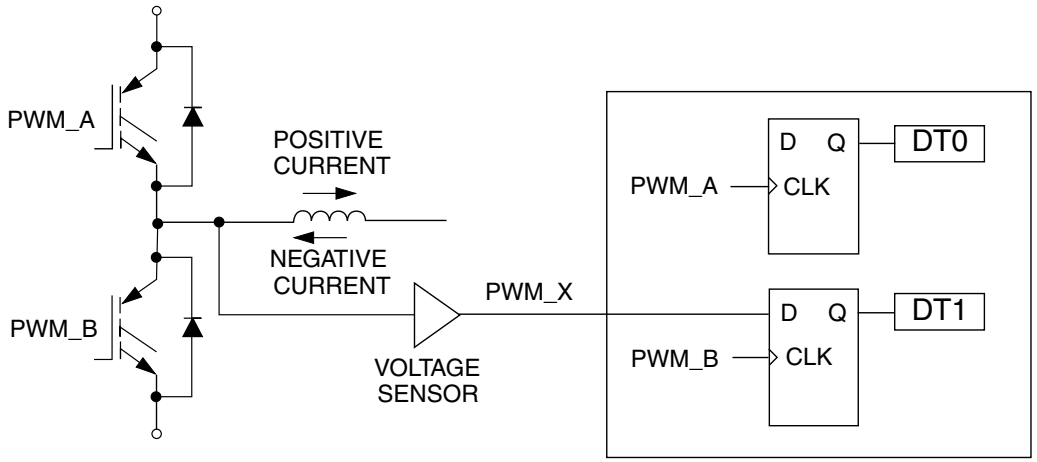


Figure 114. Current-status sense scheme for deadtime correction

Both D flip-flops latch low, CTRL[DT] = 00, during deadtime periods if the current is large and flowing out of the complementary circuit. See the preceding figure. Both D flip-flops latch the high, CTRL[DT] = 11, during deadtime periods if the current is also large and flowing into the complementary circuit.

However, under low-current, the output voltage of the complementary circuit during deadtime is between the high and low levels. The current cannot free-wheel through the opposition anti-body diode, regardless of polarity, giving additional distortion when the current crosses zero. Sampled results are CTRL[DT] = b10. Thus, the best time to change one PWM value register to another is just before the current zero crossing.

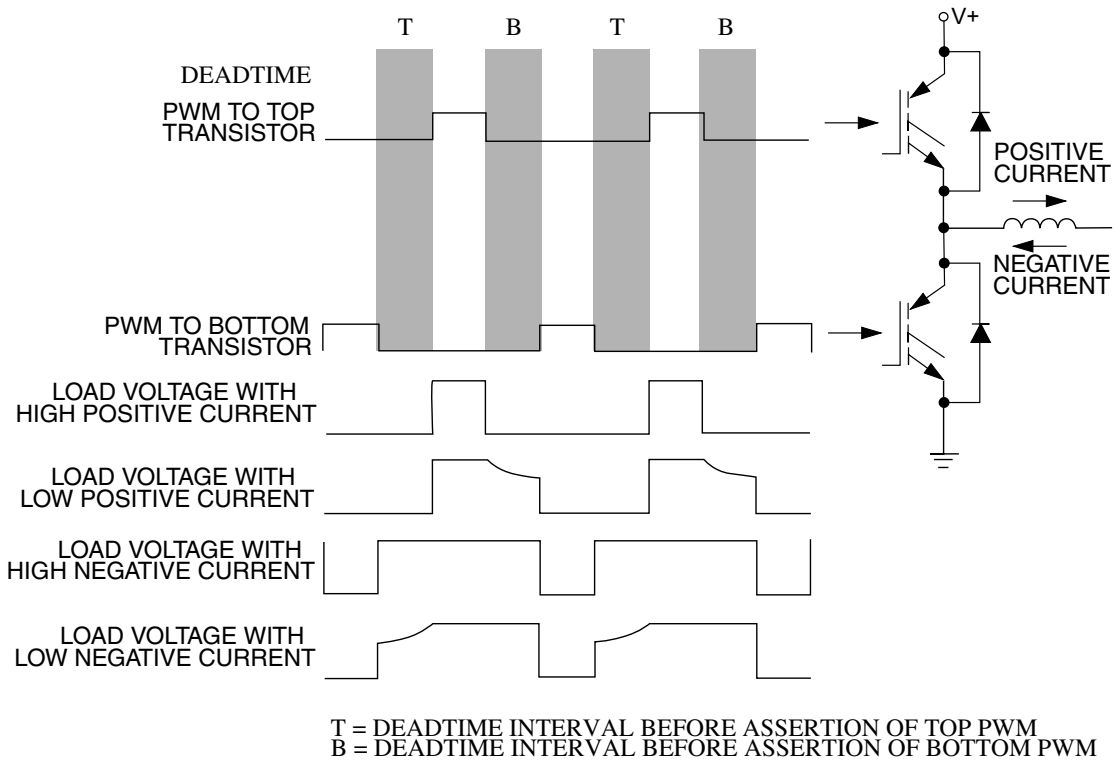


Figure 115. Output voltage waveforms

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

28.3.3.8 Output logic

The following figure shows the output logic of each submodule including how each PWM output has individual fault disabling, polarity control, and output enable. This allows for maximum flexibility when interfacing with the external circuitry.

The PWM23 and PWM45 signals which are output from the deadtime logic (as shown in Figure 116) are positive true signals. In other words, a high level on these signals should result in the corresponding transistor in the PWM inverter being turned ON. The voltage level required at the PWM output pin to turn the transistor ON or OFF is a function of the logic between the pin and the transistor. Therefore, it is imperative that the user program OCTRL[POLA] and OCTRL[POLB] before enabling the output pins. A fault condition can result in the PWM output being put in a high-impedance state, forced to a logic 1 state, or forced to a logic 0 state, depending on the values programmed into the OCTRL[PWMxFS] fields.

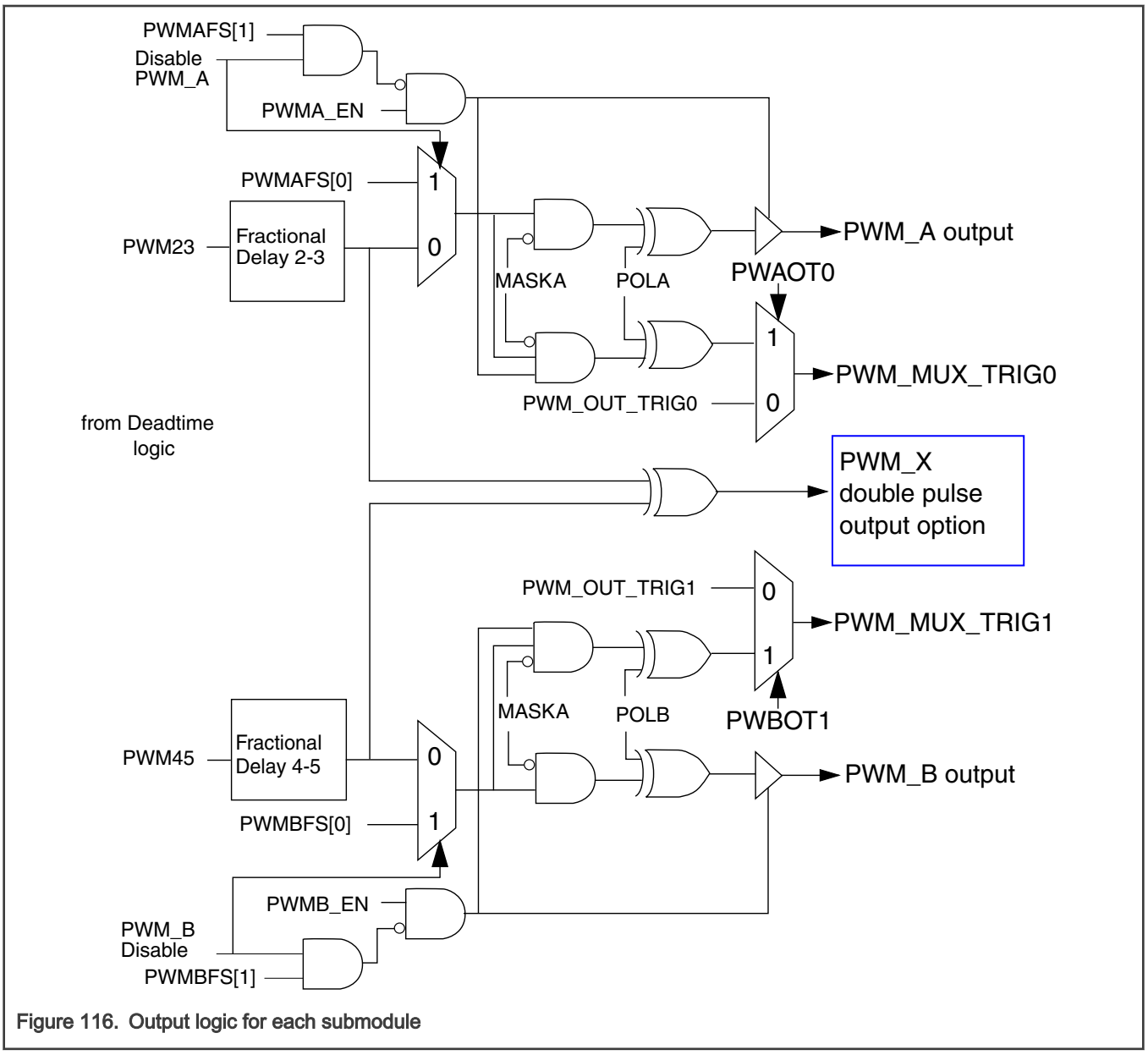


Figure 116. Output logic for each submodule

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

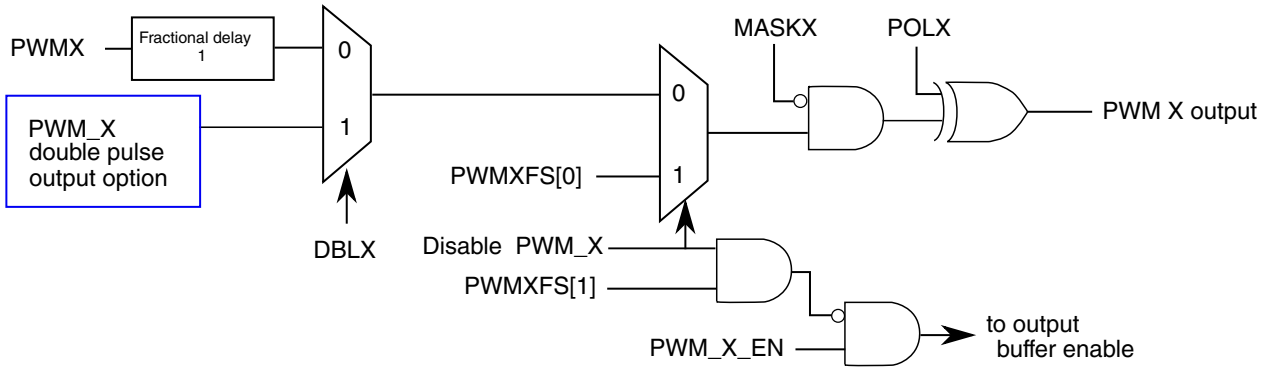


Figure 117. Output logic continued

28.3.3.9 E-Capture

The Enhanced Capture (E-Capture) logic is designed to measure both edges of an input signal. As a result, when a submodule pin is configured for input capture, the CVALx registers associated with that pin are used to record the edge values.

The following figure shows block diagram of the E-Capture circuit. Upon entering the pin input, the signal is split into two paths. One goes straight to a mux input where software can select to pass the signal directly to the capture logic for processing. The other path connects the signal to an 8-bit counter which counts both the rising and falling edges of the signal. The output of this counter is compared to an 8-bit value that is specified by the user (EDGCMPx) and when the two values are equal, the comparator generates a pulse that resets the counter. This pulse is also supplied to the mux input where software can select it to be processed by the capture logic. This feature permits the E-Capture circuit to count up to 256 edge events before initiating a capture event. This feature is useful for dividing down high frequency signals for capture processing so that capture interrupts do not overwhelm the CPU. Also, this feature can be used to generate an interrupt after "n" events have been counted.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

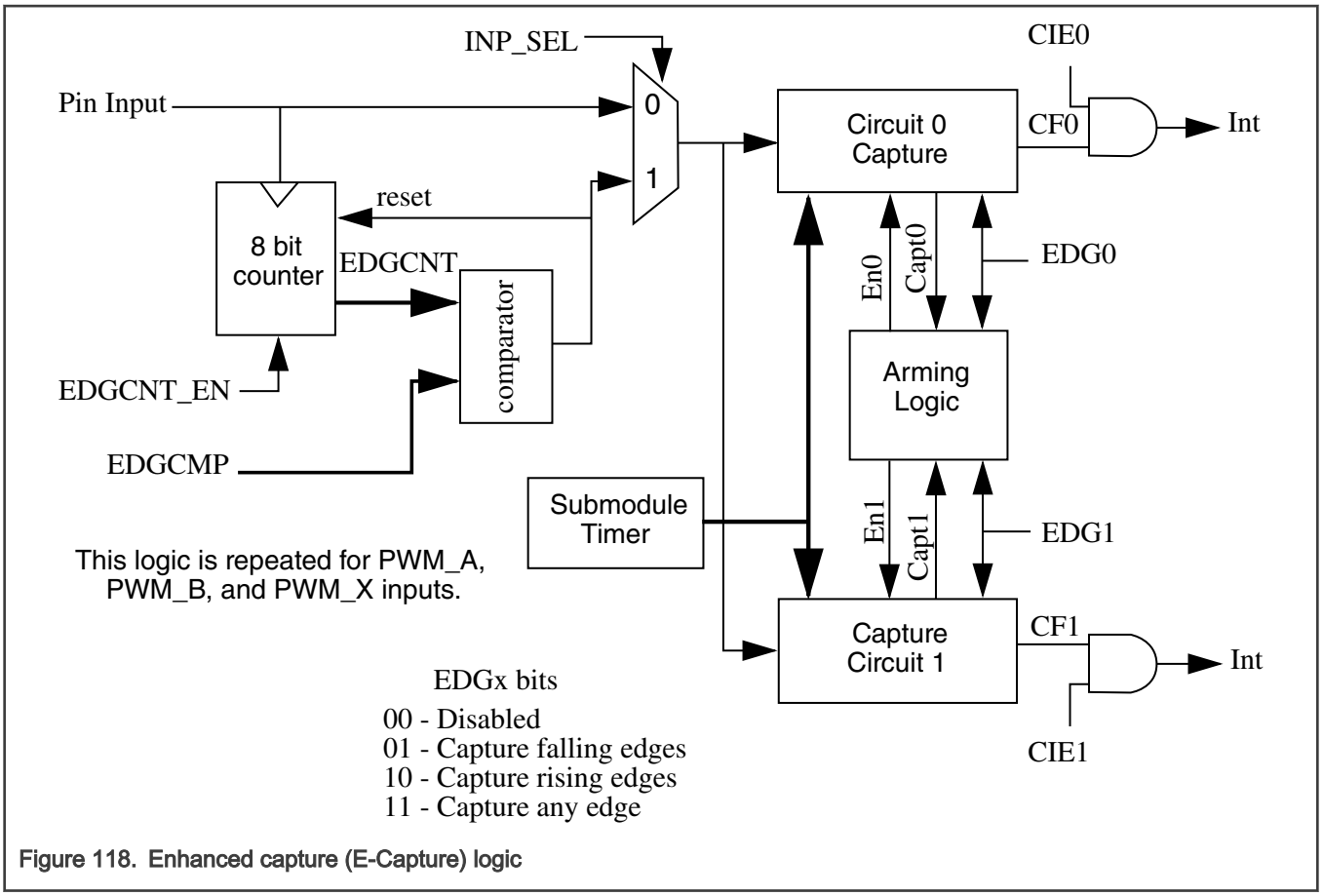


Figure 118. Enhanced capture (E-Capture) logic

Based on the mode selection, the mux selects either the pin input or the compare output from the count/compare circuit to be processed by the capture logic. The selected signal is routed to two separate capture circuits which work in tandem to capture sequential edges of the signal. The type of edge to be captured by each circuit is determined by CAPTCTRLx[EDGx1] and CAPTCTRLx[EDGx0], whose functionality is listed in the above figure. Also, controlling the operation of the capture circuits is the arming logic which allows captures to be performed in a free running (continuous) or one shot mode. In free running mode, the capture sequences will be performed indefinitely. If both capture circuits are enabled, they will work together in a ping-pong style where a capture event from one circuit leads to the arming of the other and vice versa. In one shot mode, only one capture sequence will be performed. If both capture circuits are enabled, capture circuit 0 is first armed and when a capture event occurs, capture circuit 1 is armed. Once the second capture occurs, further captures are disabled until another capture sequence is initiated. Both capture circuits are also capable of generating an interrupt to the CPU.

28.3.3.10 Fault protection

Fault protection can control any combination of PWM output pins. Faults are generated by a logic one on any of the FAULTx pins. This polarity can be changed via FCTRL[FLVL]. Each FAULTx pin can be mapped arbitrarily to any of the PWM outputs. When fault protection hardware disables PWM outputs, the PWM generator continues to run, only the output pins are forced to logic 0, logic 1, or high impedance depending on the values of OCTRL[PWMxFS].

The fault decoder disables PWM pins selected by the fault logic and the disable mapping (DISMAPn) registers. The following figure shows an example of the fault disable logic. Each bank of bits in DISMAPn control the mapping for a single PWM pin. See the following table.

The fault protection is enabled even when the module is not enabled. Therefore, a fault is latched in and must be cleared to prevent an interrupt when the module is enabled.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

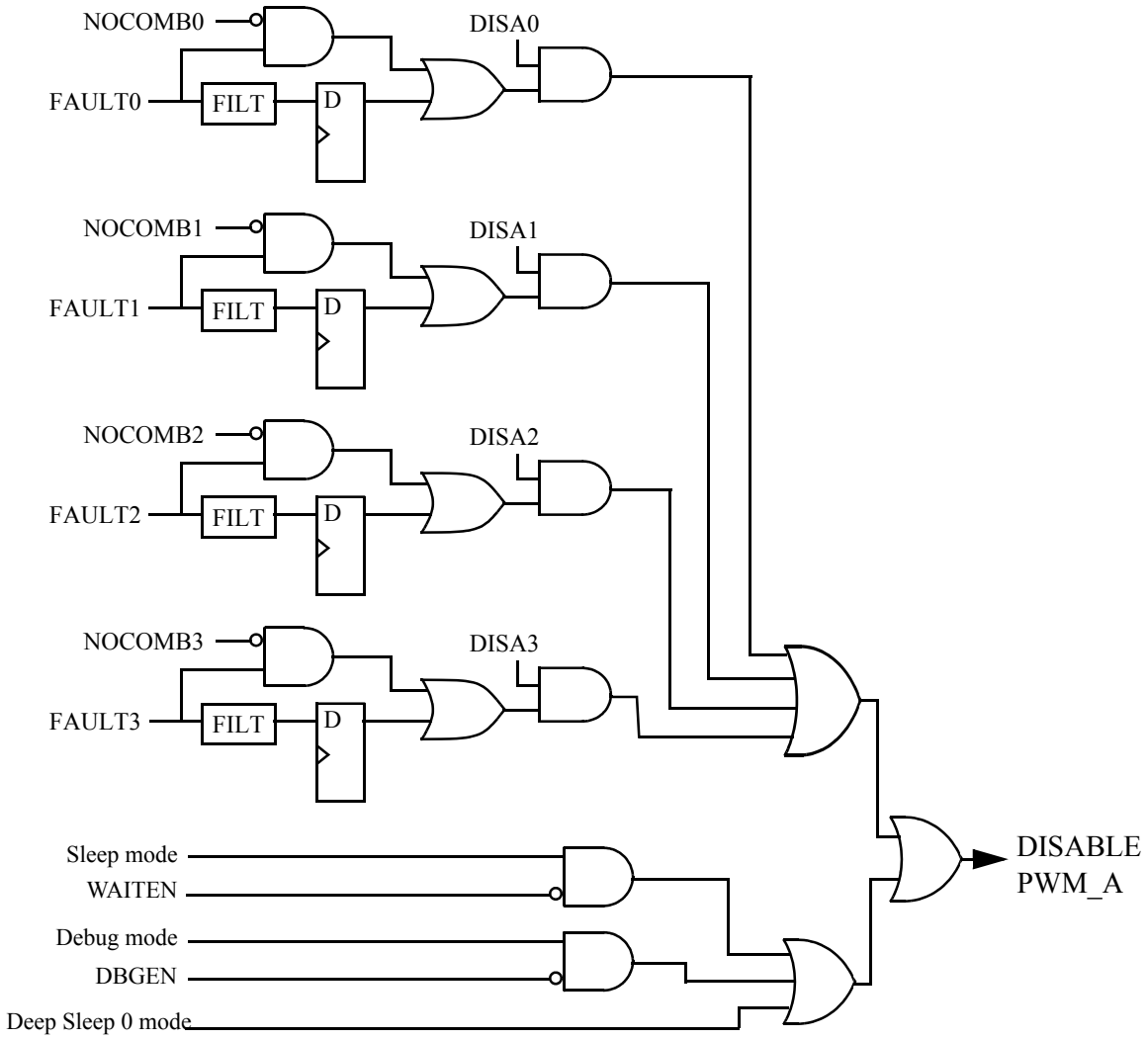


Figure 119. Fault decoder for PWM_A

Table 159. Fault Mapping

PWM Pin for 1 submodule	Controlling Register Bits
PWM_A	DISMAP0[DIS0A]
PWM_B	DISMAP0[DIS0B]
PWM_X	DISMAP0[DIS0X]

28.3.3.10.1 Fault pin filter

Each fault pin has a programmable filter that can be bypassed. The sampling period of the filter can be adjusted with FFILT[FILT_PER]. The number of consecutive samples that must agree before an input transition is recognized can be adjusted using FFILT[FILT_CNT]. Setting FFILT[FILT_PER] to all 0 disables the input filter for a given FAULTx pin.

Upon detecting a logic 0 on the filtered FAULTx pin (or a logic 1 if FCTRL[FLVLx] is set), the corresponding FSTS[FFPINx] and FSTS[FFLAGx] bits are set. FSTS[FFPINx] remains set as long as the filtered FAULTx pin is zero. Clear FSTS[FFLAGx] by writing a logic 1 to FSTS[FFLAGx].

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

If the FIEx, FAULTx pin interrupt enable bit is set, FSTS[FFLAGx] generates a CPU interrupt request. The interrupt request latch remains set until:

- Software clears FSTS[FFLAGx] by writing a logic one to the bit
- Software clears the FIEx bit by writing a logic zero to it
- A reset occurs

Even with the filter enabled, there is a combinational path from the FAULTx inputs to the PWM pins, which in turn bypasses the filter when FCTRL20[NOCOMBx] = 0. This logic is also capable of holding a fault condition in the event of loss of clock to the eFlexPWM module.

28.3.3.10.2 Automatic fault clearing

Setting an automatic clearing mode bit, FCTRL[FAUTOx] configures faults from the FAULTx pin for automatic clearing.

When FCTRL[FAUTOx] is set, disabled PWM pins are enabled when the FAULTx pin returns to logic one and a new PWM full or half cycle begins. See the following figure. If FSTS[FFULLx] is set and the fault condition on FAULTx disappears, then the disabled PWM pins are enabled at the start of next full cycle. If FSTS[FHALFx] is set, then the disabled PWM pins are enabled at the start of a half cycle. Clearing FSTS[FFLAGx] does not affect disabled PWM pins when FCTRL[FAUTOx] is set.

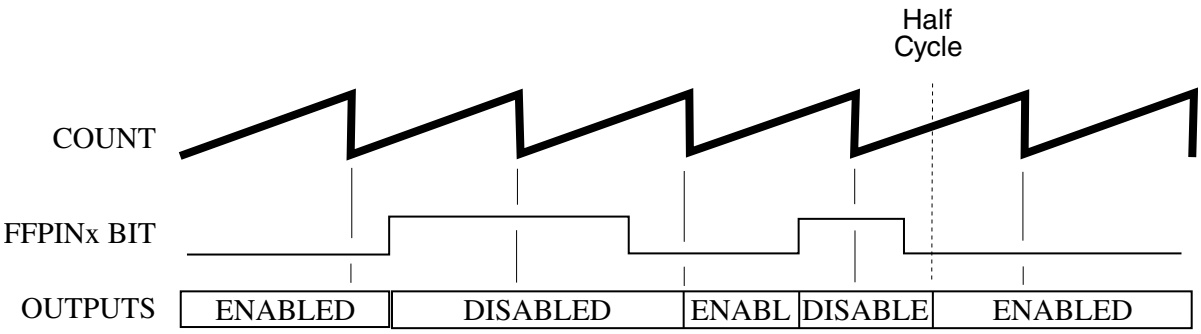


Figure 120. Automatic fault clearing

28.3.3.10.3 Manual fault clearing

Clearing the automatic clearing mode bit, FCTRL[FAUTOx], configures faults from the FAULTx pin for manual clearing:

- If the fault safety mode bits FCTRL[FSAFEx] are clear, then PWM pins disabled by the FAULTx pins are enabled when:
  - Software clears the corresponding FSTS[FFLAGx] flag
  - The pins are enabled when the next PWM full or half cycle begins regardless of the logic level detected by the filter at the FAULTx pin, as shown in Figure 121. If FSTS[FFULLx] is set, then the disabled PWM pins are enabled at the start of a full cycle. If FSTS[FHALFx] is set, then the disabled PWM pins are enabled at the start of a half cycle.
- If the fault safety mode bits FCTRL[FSAFEx] are set, then PWM pins disabled by the FAULTx pins are enabled when:
  - Software clears the corresponding FSTS[FFLAGx] flag
  - The filter detects a logic zero on the FAULTx pin at the start of the next PWM full or half cycle boundary, as shown in Figure 122. If FSTS[FFULLx] is set, then the disabled PWM pins are enabled at the start of a full cycle. If FSTS[FHALFx] is set, then the disabled PWM pins are enabled at the start of a half cycle.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

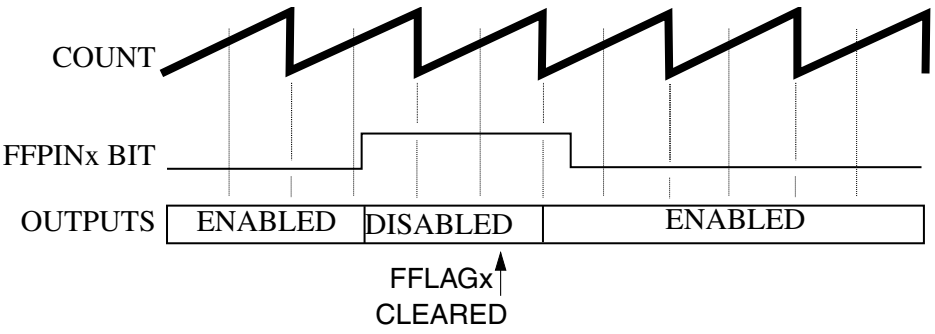


Figure 121. Manual fault clearing (FCTRL[FSAFEx] = 0, FSTS[FFULLx] = 1)

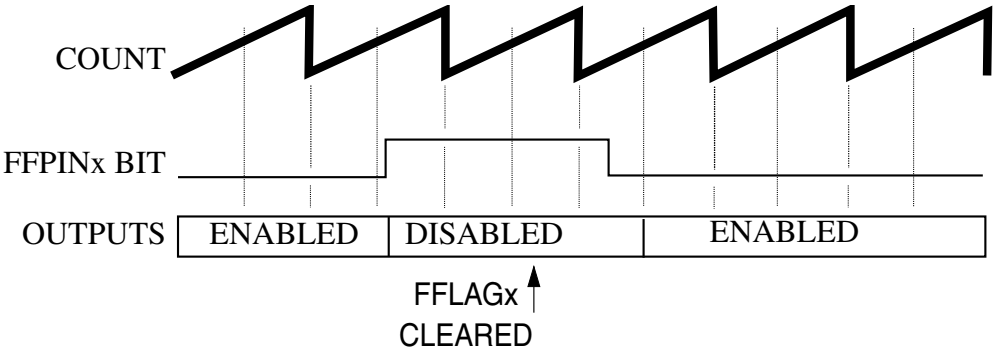


Figure 122. Manual fault clearing (FCTRL[FSAFEx] = 1, FSTS[FHALFx] = 1)

NOTE

Fault protection also applies during software output control when the SEL23 and SEL45 fields are set to select OUT23 and OUT45 bits or PWM_EXTA. Fault clearing still occurs at half or full PWM cycle boundaries while the PWM generator is engaged, MCTRL[RUN] equals one. But the OUTx bits can control the PWM pins while the PWM generator is off, MCTRL[RUN] equals zero. Thus, fault clearing occurs at IPBus cycles while the PWM generator is off and at the start of PWM cycles when the generator is engaged.

28.3.3.10.4 Fault testing

FTST[FTEST] is used to simulate a fault condition on each of the fault inputs within that fault channel.

28.3.3.11 PWM generator loading

28.3.3.11.1 Load enable

MCTRL[LDOK] enables loading of the following PWM generator parameters.

- The prescaler divisor: from CTRL[PRSC]
- The PWM period and pulse width: from the INIT, FRACVALx, and VALx registers

MCTRL[LDOK] allows the software to finish calculating all of these PWM parameters so they can be synchronously updated. The CTRL[PRSC], INIT, and VALx registers are loaded by software into a set of outer buffers. When MCTRL[LDOK] is set, these values are transferred to an inner set of registers at the beginning of the next PWM reload cycle to be used by the PWM generator when CTRL[LDMOD] is cleared. These values can be transferred to the inner set of registers immediately upon setting MCTRL[LDOK] if CTRL[LDMOD] is set. Set MCTRL[LDOK] by reading it when it is a logic zero and then writing a logic one to it. After loading, MCTRL[LDOK] is automatically cleared.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

28.3.3.11.2 Load frequency

CTRL[LDFQ] selects an integral loading frequency of one to 16 PWM reload opportunities. CTRL[LDFQ] takes effect at every PWM reload opportunity, regardless of the state of MCTRL[LDOK]. CTRL[HALF] and CTRL[FULL] control reload timing. If CTRL[FULL] is set, a reload opportunity occurs at the end of every PWM cycle when the count equals VAL1. If CTRL[HALF] is set, a reload opportunity occurs at the half cycle when the count equals VAL0. If both CTRL[HALF] and CTRL[FULL] are set, a reload opportunity occurs twice per PWM cycle when the count equals VAL1 and when it equals VAL0.

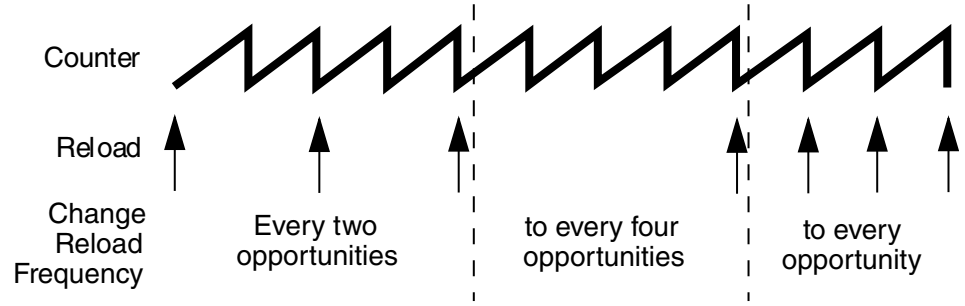


Figure 123. Full cycle reload frequency change

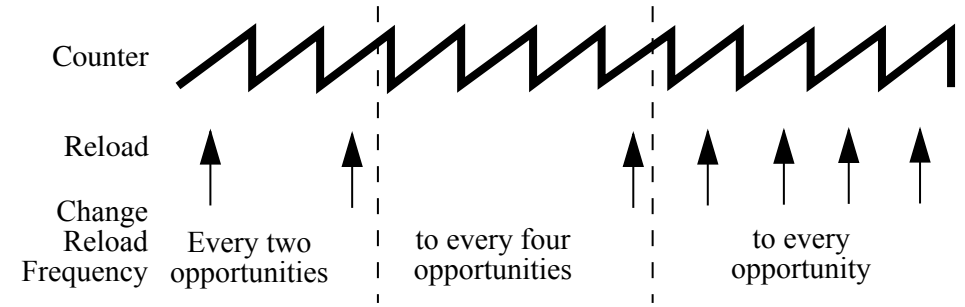


Figure 124. Half cycle reload frequency change

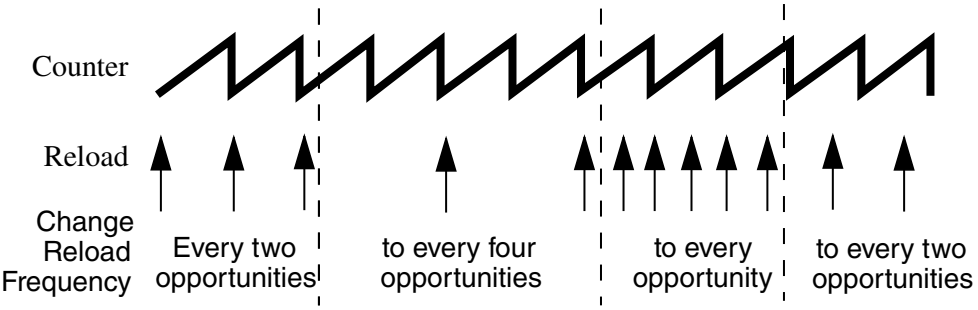


Figure 125. Full and half cycle reload frequency change

28.3.3.11.3 Reload flag

At every reload opportunity the PWM Reload Flag (STS[RF]) is set. Setting STS[RF] happens even if an actual reload is prevented by MCTRL[LDOK]. If the PWM reload interrupt enable bit INTEN[RIE] is set, the STS[RF] flag generates CPU interrupt requests allowing software to calculate new PWM parameters in real time. When INTEN[RIE] is not set, reloads still occur at the selected reload rate without generating CPU interrupt requests.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

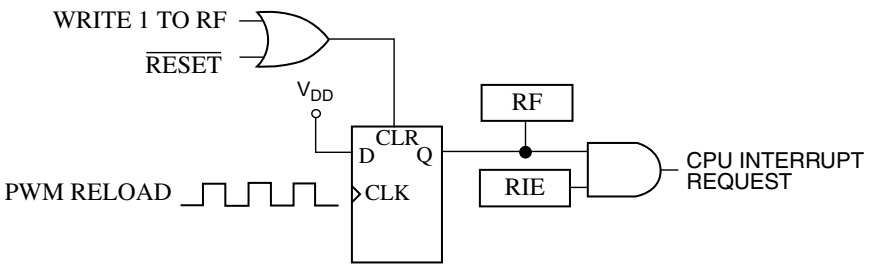


Figure 126. PWMF reload interrupt request

28.3.3.11.4 Reload errors

When one of the INIT, VALx, FRACVALx, or CTRL[PRSC] registers is updated (written by software), the STS[RUF] flag is set to indicate the data in the set of double buffered registers is not coherent. STS[RUF] is cleared by a successful reload which consists of the reload signal while MCTRL[LDOK] is set. If STS[RUF] is set and MCTRL[LDOK] is clear when the reload signal occurs, a reload error takes place and STS[REF] is set. If STS[RUF] is clear when a reload signal asserts, then the data is coherent and no error is flagged.

28.3.3.11.5 Initialization

Initialize all registers and then set MCTRL[LDOK] before setting MCTRL[RUN].

NOTE

If MCTRL[LDOK] is not set, setting MCTRL[RUN] also sets the STS[RF] flag. To prevent a CPU interrupt request, clear INTEN[RIE] before setting MCTRL[RUN].

The PWM generator uses the last values loaded if MCTRL[RUN] is cleared and then set while MCTRL[LDOK] equals zero.

When MCTRL[RUN] is cleared:

- The STS[RF] flag and pending CPU interrupt requests are not cleared
- All fault circuitry remains active
- Software/external output control remains active
- Deadtime insertion continues during software/external output control

28.3.4 Power modes

Be careful when using this module in Deep Sleep 0, Sleep, and Debug operating modes.

CAUTION

Some applications require regular software updates for proper operation. Failure to provide regular software updates could result in destroying the hardware setup.

To accommodate this situation, PWM outputs are placed in their inactive states in Deep Sleep 0 mode, and they can optionally be placed in inactive states in Sleep and Debug modes. PWM outputs are reactivated (assuming they were active beforehand) when these modes are exited.

Table 160. Modes when PWM operation is restricted

Mode	Description
Deep Sleep 0	PWM outputs are inactive.

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table 160. Modes when PWM operation is restricted (continued)

Mode	Description
Sleep	PWM outputs are driven or inactive as a function of CTRL2[WAITEN].
Debug	PWM outputs are driven or inactive as a function of CTRL2[DBGEN].

28.3.5 Clocking

Figure 127 shows the logic used to generate the main counter clock. Each submodule can select between three clock signals: the IPBus clock, EXT_CLK, and AUX_CLK. The EXT_CLK goes to all of the submodules. The AUX_CLK signal is broadcast from submodule0 and can be selected as the clock source by other submodules so that the 3-bit prescaler and MCTRL[RUN] from submodule0 can control all of the submodules.

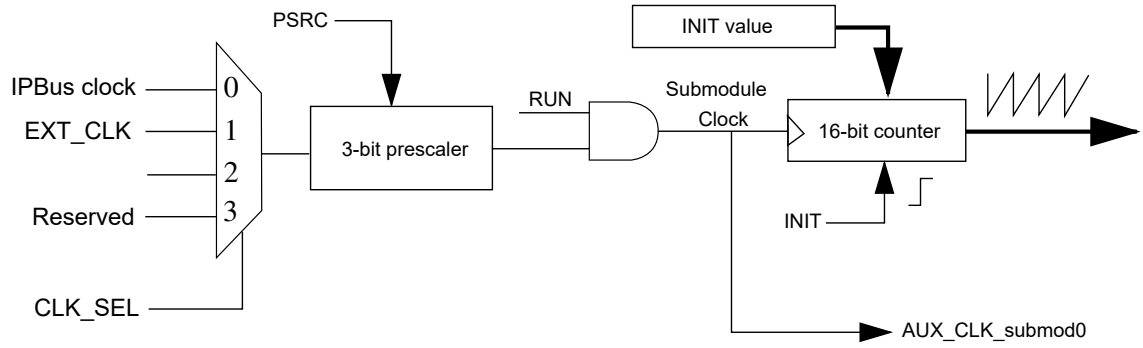


Figure 127. Clocking Block Diagram for PWM Submodule 0

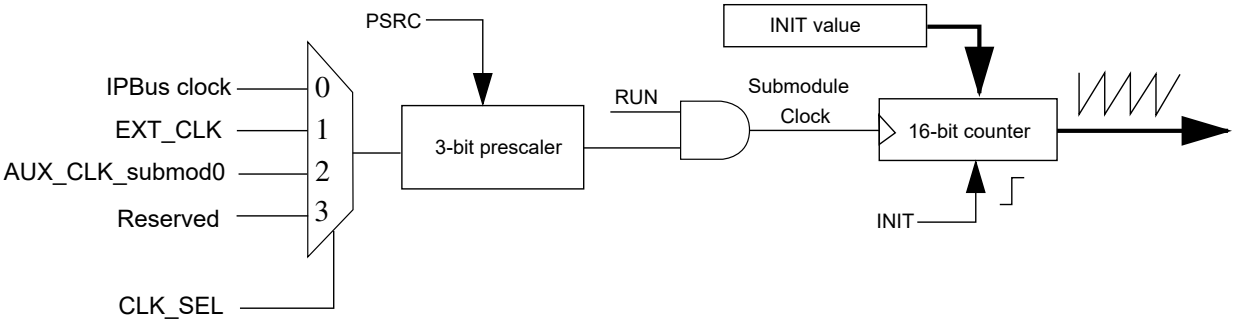


Figure 128. Clocking Block Diagram for all other PWM Submodules

To permit lower PWM frequencies, the prescaler produces the PWM clock frequency by dividing the IPBus clock frequency by 1, 2, 4, 8, 16, 32, 64, or 128. The prescaler bits, CTRL[PRSC], select the prescaler divisor. This prescaler is buffered and will not be used by the PWM generator until MCTRL[LDOK] is set and a new PWM reload cycle begins or CTRL[LDMOD] is set.

28.3.6 Resets

All eFlexPWM registers are reset to their default values upon any system reset.

The reset forces all registers to their reset states and tristates the PWM outputs.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

28.3.7 Interrupts

Each of the submodules within the eFlexPWM module can generate an interrupt from several sources. The fault logic can also generate interrupts. The interrupt service routine (ISR) must check the related interrupt enables and interrupt flags to determine the actual cause of the interrupt.

Table 161. Interrupt Summary

Core Interrupt	Interrupt Flag	Interrupt Enable	Name	Description
PWM_CMP0	SM0STS[CMPF]	SM0INTEN[CMPIE]	Submodule 0 compare interrupt	Compare event has occurred
PWM_CAP0	SM0STS[CFX1], SM0STS[CFX0]	SM0INTEN[CFX1IE], SM0INTEN[CFX0IE]	Submodule 0 input capture interrupt	Input capture event has occurred
PWM_RELOAD0	SM0STS[RF]	SM0INTEN[RIE]	Submodule 0 reload interrupt	Reload event has occurred
PWM_CMP1	SM1STS[CMPF]	SM1INTEN[CMPIE]	Submodule 1 compare interrupt	Compare event has occurred
PWM_CAP1	SM1STS[CFX1], SM1STS[CFX0]	SM1INTEN[CFX1IE], SM1INTEN[CFX0IE]	Submodule 1 input capture interrupt	Input capture event has occurred
PWM_RELOAD1	SM1STS[RF]	SM1INTEN[RIE]	Submodule 1 reload interrupt	Reload event has occurred
PWM_CMP2	SM2STS[CMPF]	SM2INTEN[CMPIE]	Submodule 2 compare interrupt	Compare event has occurred
PWM_CAP2	SM2STS[CFX1], SM2STS[CFX0]	SM2INTEN[CFX1IE], SM2INTEN[CFX0IE	Submodule 2 input capture interrupt	Input capture event has occurred
PWM_RELOAD2	SM2STS[RF]	SM2INTEN[RIE]	Submodule 2 reload interrupt	Reload event has occurred
PWM_RERR	SM0STS[REF]	SM0INTEN[REIE]	Submodule 0 reload error interrupt	Reload error has occurred
	SM1STS[REF]	SM1INTEN[REIE]	Submodule 1 reload error interrupt	
	SM2STS[REF]	SM2INTEN[REIE]	Submodule 2 reload error interrupt	
PWM_FAULT	FSTS[FFLAG]	FCTRL[FIE]	Fault input interrupt	Fault condition has been detected

28.3.8 DMA

Each submodule can request a DMA read access for its capture FIFOs and a DMA write request for its double buffered registers.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table 162. DMA summary

DMA request	DMA enable	Name	Description
Submodule 0 read request	SM0DMAEN[CX0DE]	SM0 Capture FIFO X0 read request	SM0CVAL0 contains a value to be read
Submodule 0 read request	SM0DMAEN[CX1DE]	SM0 Capture FIFO X1 read request	SM0CVAL1 contains a value to be read
Submodule 0 read request	SM0DMAEN[CAPTDE]	SM0 Capture FIFO read request source select	Selects source of submodule0 read DMA request
Submodule 0 write request	SM0DMAEN[VALDE]	SM0VALx write request	SM0VALx and SM0FRACVALx registers need to be updated
Submodule 1 read request	SM1DMAEN[CX0DE]	SM1 Capture FIFO X0 read request	SM1CVAL0 contains a value to be read
Submodule 1 read request	SM1DMAEN[CX1DE]	SM1 Capture FIFO X1 read request	SM1CVAL1 contains a value to be read
Submodule 1 read request	SM1DMAEN[CAPTDE]	SM1 Capture FIFO read request source select	Selects source of submodule1 read DMA request
Submodule 1 write request	SM1DMAEN[VALDE]	SM1VALx write request	SM1VALx and SM1FRACVALx registers need to be updated
Submodule 2 read request	SM2DMAEN[CX0DE]	SM2 Capture FIFO X0 read request	SM2CVAL0 contains a value to be read
Submodule 2 read request	SM2DMAEN[CX1DE]	SM2 Capture FIFO X1 read request	SM2CVAL1 contains a value to be read
Submodule 2 read request	SM2DMAEN[CAPTDE]	SM2 Capture FIFO read request source select	Selects source of submodule2 read DMA request
Submodule 2 write request	SM2DMAEN[VALDE]	SM2VALx write request	SM2VALx and SM2FRACVALx registers need to be updated

28.4 External signals

The PWM has pins named PWM_An, PWM_Bn, PWM_Xn, FAULTn, EXT_SYNC, EXT_FORCE, and PWMn_EXTA. The PWM also has an on-chip input called EXT_CLK and output signals called PWMn_OUT_TRIGx and PWMn_MUX_TRIGx.

28.4.1 PWM_An and PWM_Bn - External PWM output pair

These pins are the output pins of the PWM channels. These pins can be independent PWM signals or a complementary pair. When not needed as an output, they can be used as inputs to the input capture circuitry.

28.4.2 PWM_Xn - Auxiliary PWM output signal

These pins are the auxiliary output pins of the PWM channels. They can be independent PWM signals. When not needed as an output, they can be used as inputs to the input capture circuitry or used to detect the polarity of the current flowing through the complementary circuit at deadtime correction.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

28.4.3 FAULTn - Fault Inputs

These are input pins for disabling selected PWM outputs.

28.4.4 EXT_SYNC - External synchronization signal

These input signals allow a source external to the PWM to initialize the PWM counter. Therefore, the PWM can be synchronized to external circuitry.

28.4.5 EXT_FORCE - External output force signal

This input signal allows a source external to the PWM to force an update of the PWM outputs. Therefore, the PWM can be synchronized to external circuitry.

28.4.6 PWMn_EXTA - Alternate PWM control signals

These pins allow an alternate source to control the PWM_An and PWM_Bn outputs. Typically, the PWMn_EXTA input (depending on the state of MCTRL[IPOL]) is used for the generation of a complementary pair. Typical control signals include ADC conversion high/low limits, TMR outputs, GPIO inputs, and comparator outputs.

For pin input details, see chip-specific eFlexPWM information.

28.4.7 PWMn_OUT_TRIG0 and PWMn_OUT_TRIG1 - Output triggers

These outputs allow the PWM submodules to control timing of ADC conversions. See the description of the SMnTCTRL[OUT_TRIG_EN] for information about how to enable these outputs and how the compare registers match up to the output triggers.

28.4.8 PWM[n]_MUX_TRIG0 and PWM[n]_MUX_TRIG1 - Output triggers

These outputs can be either PWMn_OUT_TRIG0/PWMn_OUT_TRIG1 signals or PWM_A/PWM_B signals from output logic. See the description of the PWAOT0 and PWBOT1 bits in the Output Trigger Control Register for information about how to enable these outputs.

28.4.9 EXT_CLK - External clock signal

This signal allows a source external to the PWM (typically a timer or an off-chip source) to control the PWM clocking. Therefore, the PWM can be synchronized to the timer, or multiple chips can be synchronized to each other.

28.5 eFlexPWM register descriptions

The address of a register is the sum of a base address and an address offset. The base address is defined at the core level, and the address offset is defined at the module level. The eFlexPWM module has a set of registers for each PWM submodule, for the configuration logic, and for each fault channel.

NOTE

While the registers are 16-bit wide, they can be accessed in pairs as 32-bit registers, if the pairs are word-aligned.

Submodule registers are repeated for each PWM submodule. To designate which submodule that they are in, register names are prefixed with SMx(x=0,1...). The base address of submodule 0 is the same as the base address for the eFlexPWM module as a whole. The base address of submodule 1 is offset 0x60 from the base address for the eFlexPWM module as a whole. This 0x60 offset is based on the number of registers in a submodule. The base address of submodule 2 is equal to the base address of submodule 1 plus this same 0x60 offset.

The base address of the configuration registers is equal to the base address of the eFlexPWM module as a whole plus an offset of 0x180.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

28.5.1 eFlexPWM memory map

FlexPWM0 base address: 400A_9000h

Offset	Register	Width (In bits)	Access	Reset value
0h	Counter Register (SM0CNT)	16	R	0000h
2h	Initial Count Register (SM0INIT)	16	RW	0000h
4h	Control 2 Register (SM0CTRL2)	16	RW	0000h
6h	Control Register (SM0CTRL)	16	RW	0400h
Ah	Value Register 0 (SM0VAL0)	16	RW	0000h
Eh	Value Register 1 (SM0VAL1)	16	RW	0000h
12h	Value Register 2 (SM0VAL2)	16	RW	0000h
16h	Value Register 3 (SM0VAL3)	16	RW	0000h
1Ah	Value Register 4 (SM0VAL4)	16	RW	0000h
1Eh	Value Register 5 (SM0VAL5)	16	RW	0000h
22h	Output Control Register (SM0OCTRL)	16	RW	0000h
24h	Status Register (SM0STS)	16	RW	0000h
26h	Interrupt Enable Register (SM0INTEN)	16	RW	0000h
28h	DMA Enable Register (SM0DMAEN)	16	RW	0000h
2Ah	Output Trigger Control Register (SM0TCTRL)	16	RW	0000h
2Ch	Fault Disable Mapping Register 0 (SM0DISMAP0)	16	RW	FFFFh
30h	Deadtime Count Register 0 (SM0DTCNT0)	16	RW	07FFh
32h	Deadtime Count Register 1 (SM0DTCNT1)	16	RW	07FFh
3Ch	Capture Control X Register (SM0CAPTCTRLX)	16	RW	0000h
3Eh	Capture Compare X Register (SM0CAPTCOMPX)	16	RW	0000h
40h	Capture Value 0 Register (SM0CVAL0)	16	R	0000h
42h	Capture Value 0 Cycle Register (SM0CVAL0CYC)	16	R	0000h
44h	Capture Value 1 Register (SM0CVAL1)	16	R	0000h
46h	Capture Value 1 Cycle Register (SM0CVAL1CYC)	16	R	0000h
5Eh	Capture PWM_X Input Filter Register (SM0CAPTFILTX)	16	RW	0000h
60h	Counter Register (SM1CNT)	16	R	0000h
62h	Initial Count Register (SM1INIT)	16	RW	0000h
64h	Control 2 Register (SM1CTRL2)	16	RW	0000h
66h	Control Register (SM1CTRL)	16	RW	0400h
6Ah	Value Register 0 (SM1VAL0)	16	RW	0000h

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Offset	Register	Width (In bits)	Access	Reset value
6Eh	Value Register 1 (SM1VAL1)	16	RW	0000h
72h	Value Register 2 (SM1VAL2)	16	RW	0000h
76h	Value Register 3 (SM1VAL3)	16	RW	0000h
7Ah	Value Register 4 (SM1VAL4)	16	RW	0000h
7Eh	Value Register 5 (SM1VAL5)	16	RW	0000h
82h	Output Control Register (SM1OCTRL)	16	RW	0000h
84h	Status Register (SM1STS)	16	RW	0000h
86h	Interrupt Enable Register (SM1INTEN)	16	RW	0000h
88h	DMA Enable Register (SM1DMAEN)	16	RW	0000h
8Ah	Output Trigger Control Register (SM1TCTRL)	16	RW	0000h
8Ch	Fault Disable Mapping Register 0 (SM1DISMAP0)	16	RW	FFFFh
90h	Deadtime Count Register 0 (SM1DTCNT0)	16	RW	07FFh
92h	Deadtime Count Register 1 (SM1DTCNT1)	16	RW	07FFh
9Ch	Capture Control X Register (SM1CAPTCTRLX)	16	RW	0000h
9Eh	Capture Compare X Register (SM1CAPTCOMPX)	16	RW	0000h
A0h	Capture Value 0 Register (SM1CVAL0)	16	R	0000h
A2h	Capture Value 0 Cycle Register (SM1CVAL0CYC)	16	R	0000h
A4h	Capture Value 1 Register (SM1CVAL1)	16	R	0000h
A6h	Capture Value 1 Cycle Register (SM1CVAL1CYC)	16	R	0000h
B8h	Phase Delay Register (SM1PHASEDLY)	16	RW	0000h
BEh	Capture PWM_X Input Filter Register (SM1CAPTFILTX)	16	RW	0000h
C0h	Counter Register (SM2CNT)	16	R	0000h
C2h	Initial Count Register (SM2INIT)	16	RW	0000h
C4h	Control 2 Register (SM2CTRL2)	16	RW	0000h
C6h	Control Register (SM2CTRL)	16	RW	0400h
CAh	Value Register 0 (SM2VAL0)	16	RW	0000h
CEh	Value Register 1 (SM2VAL1)	16	RW	0000h
D2h	Value Register 2 (SM2VAL2)	16	RW	0000h
D6h	Value Register 3 (SM2VAL3)	16	RW	0000h
DAh	Value Register 4 (SM2VAL4)	16	RW	0000h
DEh	Value Register 5 (SM2VAL5)	16	RW	0000h

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Offset	Register	Width (In bits)	Access	Reset value
E2h	Output Control Register (SM2OCTRL)	16	RW	0000h
E4h	Status Register (SM2STS)	16	RW	0000h
E6h	Interrupt Enable Register (SM2INTEN)	16	RW	0000h
E8h	DMA Enable Register (SM2DMAEN)	16	RW	0000h
EAh	Output Trigger Control Register (SM2TCTRL)	16	RW	0000h
ECh	Fault Disable Mapping Register 0 (SM2DISMAP0)	16	RW	FFFFh
F0h	Deadtime Count Register 0 (SM2DTCNT0)	16	RW	07FFh
F2h	Deadtime Count Register 1 (SM2DTCNT1)	16	RW	07FFh
FCh	Capture Control X Register (SM2CAPTCTRLX)	16	RW	0000h
FEh	Capture Compare X Register (SM2CAPTCOMPX)	16	RW	0000h
100h	Capture Value 0 Register (SM2CVAL0)	16	R	0000h
102h	Capture Value 0 Cycle Register (SM2CVAL0CYC)	16	R	0000h
104h	Capture Value 1 Register (SM2CVAL1)	16	R	0000h
106h	Capture Value 1 Cycle Register (SM2CVAL1CYC)	16	R	0000h
118h	Phase Delay Register (SM2PHASEDLY)	16	RW	0000h
11Eh	Capture PWM_X Input Filter Register (SM2CAPTFILTX)	16	RW	0000h
180h	Output Enable Register (OUTEN)	16	RW	0000h
182h	Mask Register (MASK)	16	RW	0000h
184h	Software Controlled Output Register (SWCOUT)	16	RW	0000h
186h	PWM Source Select Register (DTSRCSEL)	16	RW	0000h
188h	Master Control Register (MCTRL)	16	RW	0000h
18Ah	Master Control 2 Register (MCTRL2)	16	RW	0000h
18Ch	Fault Control Register (FCTRL0)	16	RW	0000h
18Eh	Fault Status Register (FSTS0)	16	RW	See section
190h	Fault Filter Register (FFILT0)	16	RW	0000h
192h	Fault Test Register (FTST0)	16	RW	0000h
194h	Fault Control 2 Register (FCTRL20)	16	RW	0000h

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

28.5.2 Counter Register (SM0CNT - SM2CNT)

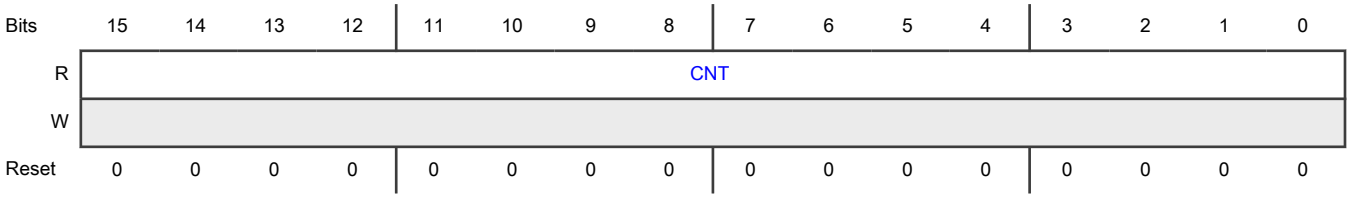
Offset

Register	Offset
SM0CNT	0h
SM1CNT	60h
SM2CNT	C0h

Function

This field displays the state of the signed 16-bit submodule counter. This register is not byte accessible. Writing this register generates bus transfer error.

Diagram



Fields

Field	Function
15-0 CNT	Counter Register Bits

28.5.3 Initial Count Register (SM0INIT - SM2INIT)

Offset

Register	Offset
SM0INIT	2h
SM1INIT	62h
SM2INIT	C2h

Function

The 16-bit signed value in this buffered register defines the initial count value for the eFlexPWM module in PWM clock periods. This is the value loaded into the submodule counter when local sync, master sync, or master reload is asserted (based on the value of CTRL2[INIT_SEL]) or when CTRL2[FORCE] is asserted and force init is enabled. For PWM operation, the buffered contents of this register are loaded into the counter at the start of every PWM cycle. This register is not byte accessible.

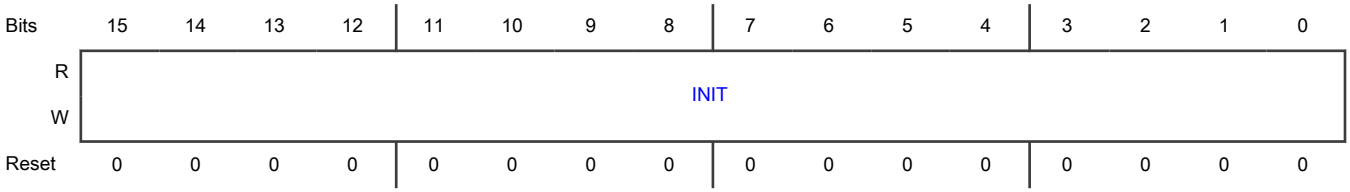
All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

NOTE

The INIT register is buffered. The value written does not take effect until MCTRL[LDOK] is set and the next PWM load cycle begins or CTRL[LDMOD] is set. This register cannot be written when MCTRL[LDOK] is set. Reading INIT reads the value in a buffer and not necessarily the value the PWM generator is currently using.

Diagram



Fields

Field	Function
15-0 INIT	Initial Count Register Bits

28.5.4 Control 2 Register (SM0CTRL2 - SM2CTRL2)

Offset

Register	Offset
SM0CTRL2	4h
SM1CTRL2	64h
SM2CTRL2	C4h

Function

Contains control fields for clock select and forcing output and initialization.This register is write-protected by MCTRL2[WRPROT] bits.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DBGEN	WAITEN	INDEP	PWM23_I...	PWM45_I...	PWMX_IN...	INIT_SEL		FRCEN	0	FORCE_SEL			RELOAD_...	CLK_SEL	
W										FORCE						
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
15 DBGEN	Debug Enable When set to one, the PWM continues to run while the chip is in Debug mode. If the device enters Debug mode and this bit is zero, then the PWM outputs are disabled until Debug mode is exited. At that point, the PWM pins resume operation as programmed in the PWM registers.
14 WAITEN	Sleep Enable When set to one, the PWM continues to run while the chip is in Sleep mode. In this mode, the peripheral clock continues to run but the CPU clock does not. If the device enters Sleep mode and this bit is zero, then the PWM outputs are disabled until Sleep mode is exited. At that point the PWM pins resume operation as programmed in the PWM registers.
13 INDEP	Independent or Complementary Pair Operation This bit determines if the PWM_A and PWM_B channels are independent PWMs or a complementary PWM pair. 0b - PWM_A and PWM_B form a complementary PWM pair. 1b - PWM_A and PWM_B outputs are independent PWMs.
12 PWM23_INIT	PWM23 Initial Value This bit determines the initial value for PWM23 and the value to which it is forced when FORCE_INIT is asserted.
11 PWM45_INIT	PWM45 Initial Value This bit determines the initial value for PWM45 and the value to which it is forced when FORCE_INIT is asserted.
10 PWMX_INIT	PWM_X Initial Value This bit determines the initial value for PWM_X and the value to which it is forced when FORCE_INIT is asserted.
9-8 INIT_SEL	Initialization Control Select These bits control the source of the INIT signal which goes to the counter. 00b - Local sync (PWM_X) causes initialization.

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	01b - Master reload from submodule 0 causes initialization. This setting should not be used in submodule 0 as it forces the INIT signal to logic 0. The submodule counter will only re-initialize when a master reload occurs. 10b - Master sync from submodule 0 causes initialization. This setting should not be used in submodule 0 as it forces the INIT signal to logic 0. 11b - EXT_SYNC causes initialization.
7 FRCEN	Force Enable This bit allows the CTRL2[FORCE] signal to initialize the counter without regard to the signal selected by CTRL2[INIT_SEL]. This is a software controlled initialization. A forced initialization will also assert the register reload if MCTRL[LDOK] is set. 0b - Initialization from a FORCE_OUT is disabled. 1b - Initialization from a FORCE_OUT is enabled.
6 FORCE	Force Initialization If CTRL2[FORCE_SEL] is set to 000, writing a 1 to this bit results in a FORCE_OUT event. This causes the following actions to be taken: • The PWM_A and PWM_B output pins assume values based on DTSRCSEL[SMxSEL23] and DTSRCSEL[SMxSEL45]. • If CTRL2[FRCEN] is set, the counter value is initialized with the INIT register value only when the submodule MCTRL[RUN] = 1 or CTRL2[CLK_SEL] = 2.
5-3 FORCE_SEL	Force Select This bit determines the source of the FORCE OUTPUT signal for this submodule. 000b - The local force signal, CTRL2[FORCE], from this submodule is used to force updates. 001b - The master force signal from submodule 0 is used to force updates. This setting should not be used in submodule 0 as it holds the FORCE OUTPUT signal to logic 0. 010b - The local reload signal from this submodule is used to force updates without regard to the state of LDOK. 011b - The master reload signal from submodule0 is used to force updates if LDOK is set. This setting should not be used in submodule0 as it holds the FORCE OUTPUT signal to logic 0. 100b - The local sync signal from this submodule is used to force updates. 101b - The master sync signal from submodule0 is used to force updates. This setting should not be used in submodule0 as it holds the FORCE OUTPUT signal to logic 0. 110b - The external force signal, EXT_FORCE, from outside the eFlexPWM module causes updates. 111b - The external sync signal, EXT_SYNC, from outside the eFlexPWM module causes updates.
2 RELOAD_SEL	Reload Source Select

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	This bit determines the source of the RELOAD signal for this submodule. When this bit is set, MCTRL[LDOK[0]] for submodule 0 should be used since the local MCTRL[LDOK] will be ignored. 0b - The local RELOAD signal is used to reload registers. 1b - The master RELOAD signal (from submodule 0) is used to reload registers. This setting should not be used in submodule 0 as it forces the RELOAD signal to logic 0.
1-0 CLK_SEL	Clock Source Select These bits determine the source of the clock signal for this submodule. 00b - The IPBus clock is used as the clock for the local prescaler and counter. 01b - EXT_CLK is used as the clock for the local prescaler and counter. 10b - Submodule 0's clock (AUX_CLK) is used as the source clock for the local prescaler and counter. This setting should not be used in submodule 0 as it forces the clock to logic 0. 11b - Reserved

28.5.5 Control Register (SM0CTRL - SM2CTRL)

Offset

Register	Offset
SM0CTRL	6h
SM1CTRL	66h
SM2CTRL	C6h

Function

Includes control settings for timing, loading, and buffering.

Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	LDFQ				HALF	FULL	DT		COMP MODE	PRSC			SPLIT	LDMOD	DBLX	DBLEN
W																
Reset	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
15-12 LDFQ	Load Frequency

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	These buffered bits select the PWM load frequency. Reset clears LDFQ, selecting loading every PWM opportunity. A PWM opportunity is determined by HALF and FULL. The register bits are write-protected by MCTRL2[WRPROT] bits. **NOTE** LDFQ takes effect when the current load cycle is complete, regardless of the state of MCTRL[LDOK]. Reading LDFQ reads the buffered values and not necessarily the values currently in effect. 0000b - Every PWM opportunity 0001b - Every 2 PWM opportunities 0010b - Every 3 PWM opportunities 0011b - Every 4 PWM opportunities 0100b - Every 5 PWM opportunities 0101b - Every 6 PWM opportunities 0110b - Every 7 PWM opportunities 0111b - Every 8 PWM opportunities 1000b - Every 9 PWM opportunities 1001b - Every 10 PWM opportunities 1010b - Every 11 PWM opportunities 1011b - Every 12 PWM opportunities 1100b - Every 13 PWM opportunities 1101b - Every 14 PWM opportunities 1110b - Every 15 PWM opportunities 1111b - Every 16 PWM opportunities
11 HALF	Half Cycle Reload This bit enables half-cycle reloads. A half cycle is defined by when the submodule counter matches the VAL0 register and does not have to be halfway through the PWM cycle. This bit is write-protected by MCTRL2[WRPROT] bits. 0b - Half-cycle reloads disabled. 1b - Half-cycle reloads enabled.
10 FULL	Full Cycle Reload This bit enables full-cycle reloads. A full cycle is defined by when the submodule counter matches the VAL1 register. Either CTRL[HALF] or CTRL[FULL] must be set to move the buffered data into the registers used by the PWM generators or CTRL[LDMOD] must be set. If both CTRL[HALF] and CTRL[FULL] are set, then reloads can occur twice per cycle. This bit is write-protected by MCTRL2[WRPROT] bits. 0b - Full-cycle reloads disabled.

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	1b - Full-cycle reloads enabled.
9-8 DT	Deadtime These bits reflect the sampled values of the PWM_X input at the end of each deadtime. Sampling occurs at the end of deadtime 0 for DT[0] and the end of deadtime 1 for DT[1]. Reset clears these bits. The register bits are write-protected by MCTRL2[WRPROT] bits.
7 COMPMODE	Compare Mode This bit controls how comparisons are made between the VAL* registers and the PWM submodule counter. This bit can be written one time after which it requires a reset to release the bit for writing again. 0b - The VAL* registers and the PWM counter are compared using an "equal to" method. This means that PWM edges are only produced when the counter is equal to one of the VAL* register values. This implies that a PWM_A output that is high at the end of a period maintains this state until a match with VAL3 clears the output in the following period. 1b - The VAL* registers and the PWM counter are compared using an "equal to or greater than" method. This means that PWM edges are produced when the counter is equal to or greater than one of the VAL* register values. This implies that a PWM_A output that is high at the end of a period could go low at the start of the next period if the starting counter value is greater than (but not necessarily equal to) the new VAL3 value.
6-4 PRSC	Prescaler These buffered bits select the divide ratio of the PWM clock frequency selected by CTRL2[CLK_SEL]. **NOTE** Reading CTRL[PRSC] reads the buffered values and not necessarily the values currently in effect. CTRL[PRSC] takes effect at the beginning of the next PWM cycle and only when the load okay bit MCTRL[LDOK] is set, or CTRL[LDMOD] is set. This field cannot be written when MCTRL[LDOK] is set. 000b - Prescaler 1. PWM clock frequency = f_{clk} 001b - Prescaler 2. PWM clock frequency = f_{clk} /2 010b - Prescaler 4. PWM clock frequency = f_{clk} /4 011b - Prescaler 8. PWM clock frequency = f_{clk} /8 100b - Prescaler 16. PWM clock frequency = f_{clk} /16 101b - Prescaler 32. PWM clock frequency = f_{clk} /32 110b - Prescaler 64. PWM clock frequency = f_{clk} /64 111b - Prescaler 128. PWM clock frequency = f_{clk} /128
3 SPLIT	Split the DBLPWM signal to PWM_A and PWM_B This bit is used in independent mode when DBLEN is set. This bit allows the two PWM pulses generated by DBLEN to be split with one pulse on PWM_A and one on PWM_B. The two pulses within the same PWM period are created by an XOR function of the PWM_A and PWM_B sources. The splitting function causes PWM_A to output the pulse that occurs when the PWM_A source is 1 and the PWM_B source is 0. The

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	PWM_B output occurs when the PWM_B source is 1 and the PWM_A source is 0. (See Double switching PWMs .) This bit is write-protected by MCTRL2[WRPROT] bits. 0b - DBLPWM is not split. PWM_A and PWM_B each have double pulses. 1b - DBLPWM is split to PWM_A and PWM_B.
2 LDMOD	Load Mode Select This bit selects the timing of loading the buffered registers for this submodule. This bit is write-protected by MCTRL2[WRPROT] bits. 0b - Buffered registers of this submodule are loaded and take effect at the next PWM reload if MCTRL[LDOK] is set. 1b - Buffered registers of this submodule are loaded and take effect immediately upon MCTRL[LDOK] being set. In this case, it is not necessary to set CTRL[FULL] or CTRL[HALF].
1 DBLX	PWM_X Double Switching Enable This bit enables the double switching behavior on PWM_X. When this bit is set, the PWM_X output shall be the exclusive OR combination of PWM_A and PWM_B prior to polarity and masking considerations. This bit is write-protected by MCTRL2[WRPROT] bits. 0b - PWM_X double pulse disabled. 1b - PWM_X double pulse enabled.
0 DBLEN	Double Switching Enable This bit enables the double switching PWM behavior (See Double switching PWMs). This bit is write-protected by MCTRL2[WRPROT] bits. 0b - Double switching disabled. 1b - Double switching enabled.

28.5.6 Value Register 0 (SM0VAL0 - SM2VAL0)

Offset

Register	Offset
SM0VAL0	Ah
SM1VAL0	6Ah
SM2VAL0	CAh

Function

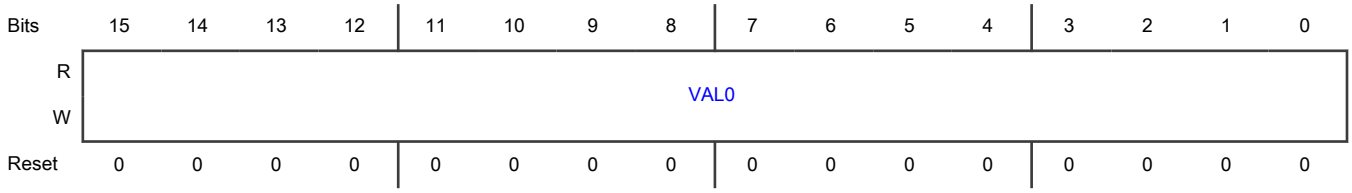
NOTE

The VAL0 register is buffered. The value written does not take effect until MCTRL[LDOK] is set and the next PWM load cycle begins or CTRL[LDMOD] is set. VAL0 cannot be written when MCTRL[LDOK] is set. Reading VAL0 reads the value in a buffer. It is not necessarily the value that the PWM generator is currently using.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram



Fields

Field	Function
15-0 VAL0	Value 0 The 16-bit signed value in this buffered register defines the mid-cycle reload point for the PWM in PWM clock periods. This value also defines when the PWM_X signal is set and the local sync signal is reset. This register is not byte accessible. **NOTE** The actual behavior takes effect when counter equal VAL0+1.

28.5.7 Value Register 1 (SM0VAL1 - SM2VAL1)

Offset

Register	Offset
SM0VAL1	Eh
SM1VAL1	6Eh
SM2VAL1	CEh

Function

NOTE

The VAL1 register is buffered. The value written does not take effect until MCTRL[LDOK] is set and the next PWM load cycle begins or CTRL[LDMOD] is set. VAL1 cannot be written when MCTRL[LDOK] is set. Reading VAL1 reads the value in a buffer. It is not necessarily the value that the PWM generator is currently using.

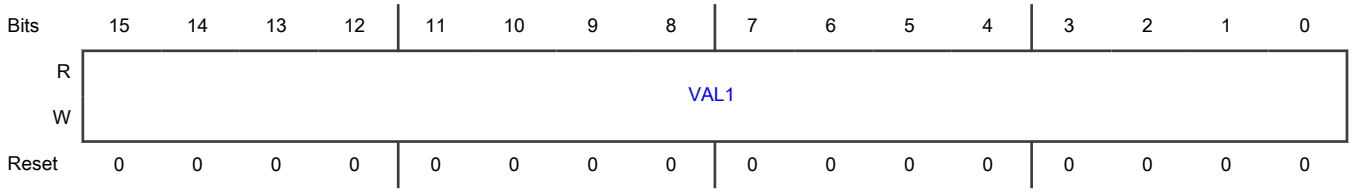
When using FRACVAL1, limit the maximum value of VAL1 to 0xFFFE for unsigned applications or to 0x7FFE for signed applications, to avoid counter rollovers caused by accumulating the fractional period defined by FRACVAL1.

If the VAL1 register defines the timer period (Local Sync is selected as the counter initialization signal), a 100% duty cycle cannot be achieved on the PWM_X output. After the count reaches VAL1, the PWM_X output is low for a minimum of one count every cycle. When the Master Sync signal (only originated by the Local Sync from submodule 0) is used to control the timer period, the VAL1 register can be free for other functions such as PWM generation without the duty cycle limitation.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram



Fields

Field	Function
15-0 VAL1	Value 1 The 16-bit signed value written to this buffered register defines the modulo count value (maximum count) for the submodule counter. When reaching this count value, the counter reloads itself with the contents of the INIT register and asserts the local sync signal while resetting PWM_X. This register is not byte accessible. **NOTE** The actual behavior takes effect when the counter equals VAL1+1.

28.5.8 Value Register 2 (SM0VAL2 - SM2VAL2)

Offset

Register	Offset
SM0VAL2	12h
SM1VAL2	72h
SM2VAL2	D2h

Function

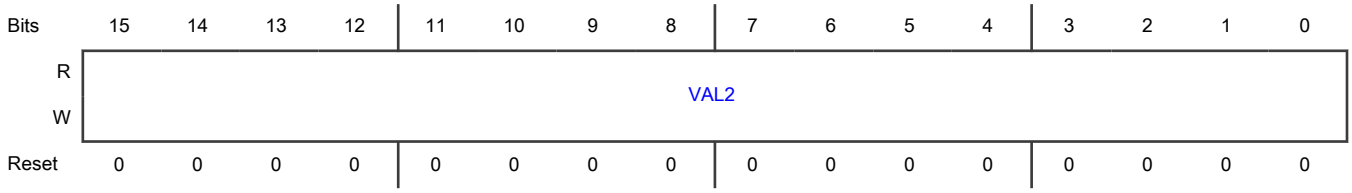
NOTE

The VAL2 register is buffered. The value written does not take effect until MCTRL[LDOK] is set and the next PWM load cycle begins or CTRL[LDMOD] is set. VAL2 cannot be written when MCTRL[LDOK] is set. Reading VAL2 reads the value in a buffer and not necessarily the value that the PWM generator is currently using.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram



Fields

Field	Function
15-0 VAL2	Value 2 The 16-bit signed value in this buffered register defines the count value to set PWM23 high. This register is not byte accessible. **NOTE** The actual behavior takes effect when the counter equals VAL2+1.

28.5.9 Value Register 3 (SM0VAL3 - SM2VAL3)

Offset

Register	Offset
SM0VAL3	16h
SM1VAL3	76h
SM2VAL3	D6h

Function

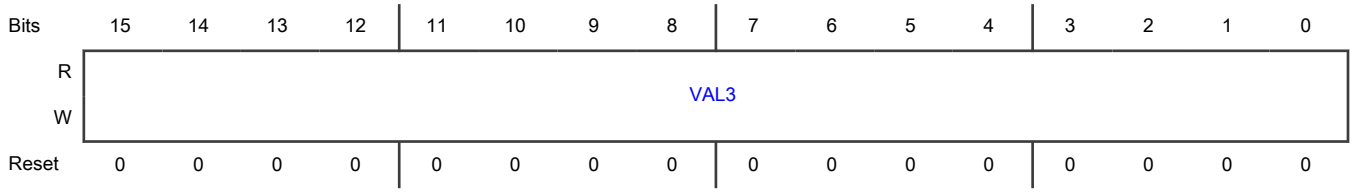
NOTE

The VAL3 register is buffered. The value written does not take effect until MCTRL[LDOK] is set and the next PWM load cycle begins or CTRL[LDMOD] is set. VAL3 cannot be written when MCTRL[LDOK] is set. Reading VAL3 reads the value in a buffer and not necessarily the value that the PWM generator is currently using.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram



Fields

Field	Function
15-0 VAL3	Value 3 The 16-bit signed value in this buffered register defines the count value to set PWM23 low. This register is not byte accessible. **NOTE** The actual behavior takes effect when the counter equals VAL3+1.

28.5.10 Value Register 4 (SM0VAL4 - SM2VAL4)

Offset

Register	Offset
SM0VAL4	1Ah
SM1VAL4	7Ah
SM2VAL4	DAh

Function

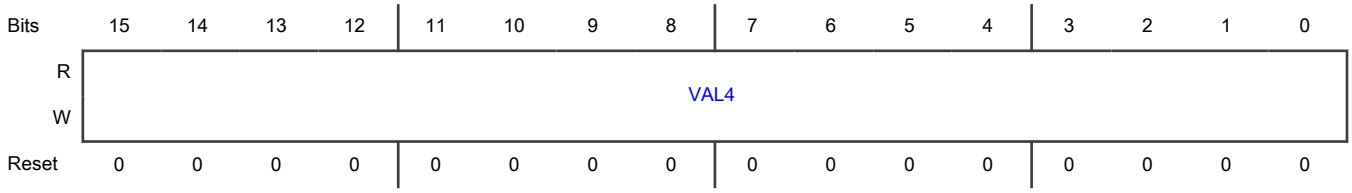
NOTE

The VAL4 register is buffered. The value written does not take effect until MCTRL[LDOK] is set and the next PWM load cycle begins or CTRL[LDMOD] is set. VAL4 cannot be written when MCTRL[LDOK] is set. Reading VAL4 reads the value in a buffer and not necessarily the value that the PWM generator is currently using.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram



Fields

Field	Function
15-0 VAL4	Value 4 The 16-bit signed value in this buffered register defines the count value to set PWM45 high. This register is not byte accessible. **NOTE** The actual behavior takes effect when the counter equals VAL4+1.

28.5.11 Value Register 5 (SM0VAL5 - SM2VAL5)

Offset

Register	Offset
SM0VAL5	1Eh
SM1VAL5	7Eh
SM2VAL5	DEh

Function

NOTE

The VAL5 register is buffered. The value written does not take effect until MCTRL[LDOK] is set and the next PWM load cycle begins or CTRL[LDMOD] is set. VAL5 cannot be written when MCTRL[LDOK] is set. Reading VAL5 reads the value in a buffer and not necessarily the value that the PWM generator is currently using.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	VAL5															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
15-0 VAL5	Value 5 The 16-bit signed value in this buffered register defines the count value to set PWM45 low. This register is not byte accessible. **NOTE** The actual behavior takes effect when the counter equals VAL5+1.

28.5.12 Output Control Register (SM0OCTRL - SM2OCTRL)

Offset

Register	Offset
SM0OCTRL	22h
SM1OCTRL	82h
SM2OCTRL	E2h

Function

Contains output controls for fault states. This register is write-protected by MCTRL2[WRPROT] bits.

Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PWMA_IN	PWMB_IN	PWMX_IN	0		POLA	POLB	POLX	0		PWMAFS		PWMBFS		PWMXFS	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
15 PWMA_IN	PWM_A Input This bit shows the logic value currently being driven into the PWM_A input. The reset state is undefined.

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
14 PWMB_IN	PWM_B Input This bit shows the logic value currently being driven into the PWM_B input. The reset state is undefined.
13 PWMX_IN	PWM_X Input This bit shows the logic value currently being driven into the PWM_X input. The reset state is undefined.
12-11 —	Reserved
10 POLA	PWM_A Output Polarity This bit inverts the PWM_A output polarity. 0b - PWM_A output not inverted. A high level on the PWM_A pin represents the "on" or "active" state. 1b - PWM_A output inverted. A low level on the PWM_A pin represents the "on" or "active" state.
9 POLB	PWM_B Output Polarity This bit inverts the PWM_B output polarity. 0b - PWM_B output not inverted. A high level on the PWM_B pin represents the "on" or "active" state. 1b - PWM_B output inverted. A low level on the PWM_B pin represents the "on" or "active" state.
8 POLX	PWM_X Output Polarity This bit inverts the PWM_X output polarity. 0b - PWM_X output not inverted. A high level on the PWM_X pin represents the "on" or "active" state. 1b - PWM_X output inverted. A low level on the PWM_X pin represents the "on" or "active" state.
7-6 —	Reserved
5-4 PWMAFS	PWM_A Fault State These bits determine the fault state for the PWM_A output during fault conditions and Deep Sleep 0 mode. It may also define the output state during Sleep and Debug modes depending on the settings of CTRL2[WAITEN] and CTRL2[DBGEN]. 00b - Output is forced to logic 0 state prior to consideration of output polarity control. 01b - Output is forced to logic 1 state prior to consideration of output polarity control. 10b,11b - Output is put in a high-impedance state.
3-2 PWMBFS	PWM_B Fault State

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	These bits determine the fault state for the PWM_B output during fault conditions and Deep Sleep 0 mode. It may also define the output state during Sleep and Debug modes depending on the settings of CTRL2[WAITEN] and CTRL2[DBGEN]. 00b - Output is forced to logic 0 state prior to consideration of output polarity control. 01b - Output is forced to logic 1 state prior to consideration of output polarity control. 10b,11b - Output is put in a high-impedance state.
1-0 PWMXFS	PWM_X Fault State These bits determine the fault state for the PWM_X output during fault conditions and Deep Sleep 0 mode. It may also define the output state during Sleep and Debug modes depending on the settings of CTRL2[WAITEN] and CTRL2[DBGEN]. 00b - Output is forced to logic 0 state prior to consideration of output polarity control. 01b - Output is forced to logic 1 state prior to consideration of output polarity control. 10b,11b - Output is put in a high-impedance state.

28.5.13 Status Register (SM0STS - SM2STS)

Offset

Register	Offset
SM0STS	24h
SM1STS	84h
SM2STS	E4h

Function

Contains Compare and Capture flag status.

Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	RUF	REF	RF	0				CFX1	CFX0	CMPF					
W			W1C	W1C					W1C	W1C	W1C					
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
15	Reserved

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
—	
14 RUF	Registers Updated Flag This bit is set when one of the INIT, VALx, FRACVALx, or CTRL[PRSC] registers has been written, which indicates potentially non-coherent data in the set of double buffered registers. Clear this bit by a proper reload sequence consisting of a reload signal while MCTRL[LDOK] = 1. Reset clears this bit. 0b - No register update has occurred since last reload. 1b - At least one of the double buffered registers has been updated since the last reload.
13 REF	Reload Error Flag This bit is set when a reload cycle occurs while MCTRL[LDOK] is 0 and the double buffered registers are in a non-coherent state (STS[RUF] = 1). Clear this bit by writing a logic one to this location. Reset clears this bit. 0b - No reload error occurred. 1b - Reload signal occurred with non-coherent data and MCTRL[LDOK] = 0.
12 RF	Reload Flag This bit is set at the beginning of every reload cycle regardless of the state of MCTRL[LDOK]. Clear this bit by writing a logic one to this location when DMAEN[VALDE] is clear (non-DMA mode). This bit can also be cleared by the DMA done signal when DMAEN[VALDE] is set (DMA mode) . Reset clears this bit. 0b - No new reload cycle since last STS[RF] clearing 1b - New reload cycle since last STS[RF] clearing
11-8 —	Reserved
7 CFX1	Capture Flag X1 This bit is set when a capture event occurs on the Capture X1 circuit . This bit is cleared by writing a one to this bit position if DMAEN[CX1DE] is clear (non-DMA mode) or by the DMA done signal if DMAEN[CX1DE] is set (DMA mode). This bit can be cleared when word count of the corresponding FIFO is less than water mark level. Reset clears this bit.
6 CFX0	Capture Flag X0 This bit is set when a capture event occurs on the Capture X0 circuit . This bit is cleared by writing a one to this bit position if DMAEN[CX0DE] is clear (non-DMA mode) or by the DMA done signal if DMAEN[CX0DE] is set (DMA mode). This bit can be cleared when word count of the corresponding FIFO is less than water mark level. Reset clears this bit.
5-0 CMPF	Compare Flags These bits are set when the submodule counter value matches the value of one of the VALx registers. Clear these bits by writing a 1 to a bit position. 00_0000b - No compare event has occurred for a particular VALx value. 00_0001b - A compare event has occurred for a particular VALx value.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

28.5.14 Interrupt Enable Register (SM0INTEN - SM2INTEN)

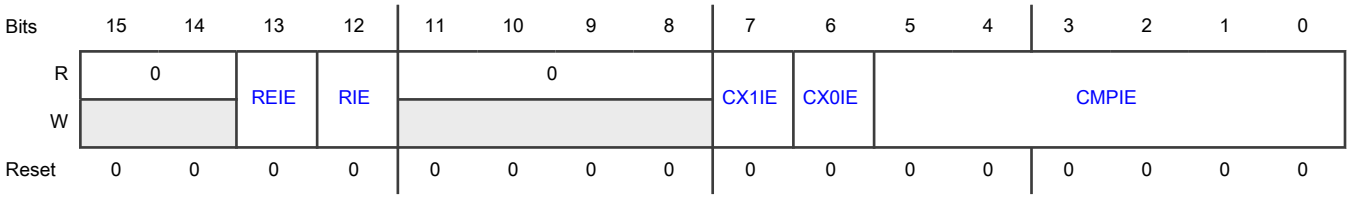
Offset

Register	Offset
SM0INTEN	26h
SM1INTEN	86h
SM2INTEN	E6h

Function

Contains Compare and Capture interrupt enables.

Diagram



Fields

Field	Function
15-14 —	Reserved
13 REIE	Reload Error Interrupt Enable This bit enables the reload error flag, STS[REF], to generate CPU interrupt requests. Reset clears this bit. 0b - STS[REF] CPU interrupt requests disabled 1b - STS[REF] CPU interrupt requests enabled
12 RIE	Reload Interrupt Enable This bit enables the reload flag, STS[RF], to generate CPU interrupt requests. Reset clears this bit. 0b - STS[RF] CPU interrupt requests disabled 1b - STS[RF] CPU interrupt requests enabled
11-8 —	Reserved
7 CX1IE	Capture X 1 Interrupt Enable This bit allows the STS[CFX1] flag to create an interrupt request to the CPU. Do not set this bit and DMAEN[CX1DE].

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	0b - Interrupt request disabled for STS[CFX1]. 1b - Interrupt request enabled for STS[CFX1].
6 CX0IE	Capture X 0 Interrupt Enable This bit allows the STS[CFX0] flag to create an interrupt request to the CPU. Do not set this bit and DMAEN[CX0DE]. 0b - Interrupt request disabled for STS[CFX0]. 1b - Interrupt request enabled for STS[CFX0].
5-0 CMPIE	Compare Interrupt Enables These bits enable the STS[CMPF] flags to cause a compare interrupt request to the CPU. 00_0000b - The corresponding STS[CMPF] bit will not cause an interrupt request. 00_0001b - The corresponding STS[CMPF] bit will cause an interrupt request.

28.5.15 DMA Enable Register (SM0DMAEN - SM2DMAEN)

Offset

Register	Offset
SM0DMAEN	28h
SM1DMAEN	88h
SM2DMAEN	E8h

Function

Contains controls for DMA. This register is write-protected by MCTRL2[WRPROT] bits.

Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0						VALDE	FAND	CAPTDE		0				CX1DE	CX0DE
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
15-10	Reserved

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
—	
9 VALDE	Value Registers DMA Enable This bit enables DMA write requests for the VALx and FRACVALx registers when STS[RF] is set. Reset clears this bit. 0b - DMA write requests disabled 1b - Enabled. DMA write requests for the VALx and FRACVALx registers enabled
8 FAND	FIFO Watermark AND Control This bit works in conjunction with the DMAEN[CAPTDE] field when it is set to watermark mode (DMAEN[CAPTDE] = 01). While DMAEN[CAxDE], DMAEN[CBxDE], and DMAEN[CXxDE] determine which FIFO watermarks the DMA read request is sensitive to. This bit determines if the selected watermarks are AND'ed together or OR'ed together to create the request. 0b - Selected FIFO watermarks are OR'ed together. 1b - Selected FIFO watermarks are AND'ed together.
7-6 CAPTDE	Capture DMA Enable Source Select These bits select the source of enabling the DMA read requests for the capture FIFOs. Reset clears these bits. 00b - Read DMA requests disabled. 01b - Exceeding a FIFO watermark sets the DMA read request. This requires at least one of DMAEN[CX1DE], or DMAEN[CX0DE] to be set to determine which watermark(s) the DMA request is sensitive. 10b - A local synchronization (VAL1 matches counter) sets the read DMA request. 11b - A local reload (STS[RF] being set) sets the read DMA request.
5-2 —	Reserved
1 CX1DE	Capture X1 FIFO DMA Enable This bit enables DMA read requests for the Capture X1 FIFO data when STS[CFX1] is set. Reset clears this bit. Do not set both this bit and INTEN[CX1IE].
0 CX0DE	Capture X0 FIFO DMA Enable This bit enables DMA read requests for the Capture X0 FIFO data when STS[CFX0] is set. Reset clears this bit. Do not set both this bit and INTEN[CX0IE].

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

28.5.16 Output Trigger Control Register (SM0TCTRL - SM2TCTRL)

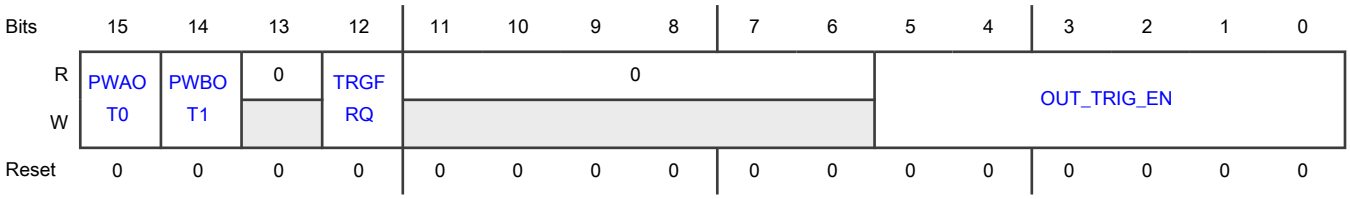
Offset

Register	Offset
SM0TCTRL	2Ah
SM1TCTRL	8Ah
SM2TCTRL	EAh

Function

Contains trigger controls.This register is write-protected by MCTRL2[WRPROT] bits.

Diagram



Fields

Field	Function
15 PWAOT0	Mux Output Trigger 0 Source Select This bit selects which signal to bring out on the PWM's PWM_MUX_TRIG0 port. 0b - Route the PWM_OUT_TRIG0 signal to PWM_MUX_TRIG0 port. 1b - Route the PWM_A output to the PWM_MUX_TRIG0 port.
14 PWBOT1	Mux Output Trigger 1 Source Select This bit selects which signal to bring out on the PWM's PWM_MUX_TRIG1 port. 0b - Route the PWM_OUT_TRIG1 signal to PWM_MUX_TRIG1 port. 1b - Route the PWM_B output to the PWM_MUX_TRIG1 port.
13 —	Reserved
12 TRGFRQ	Trigger Frequency This bit allows control over the frequency of the trigger outputs when using non-zero values of CTRL[LDFQ]. 0b - Trigger outputs are generated during every PWM period even if the PWM is not reloaded every period due to CTRL[LDFQ] being non-zero. 1b - Trigger outputs are generated only during the final PWM period prior to a reload opportunity when the PWM is not reloaded every period due to CTRL[LDFQ] being non-zero.

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
11-6 —	Reserved
5-0 OUT_TRIG_EN	Output Trigger Enables These bits enable the generation of PWM_OUT_TRIG0 and PWM_OUT_TRIG1 outputs based on the counter value matching the value in one or more of the VAL0-5 registers. **NOTE** Due to delays in creating the PWM outputs, the output trigger signals lead the PWM output edges by 2-3 clock cycles depending on the fractional cycle value being used. 1x_xxxxb - PWM_OUT_TRIG1 will set when the counter value matches the VAL5 value. x1_xxxxb - PWM_OUT_TRIG0 will set when the counter value matches the VAL4 value. xx_1xxxb - PWM_OUT_TRIG1 will set when the counter value matches the VAL3 value. xx_x1xxb - PWM_OUT_TRIG0 will set when the counter value matches the VAL2 value. xx_xx1xb - PWM_OUT_TRIG1 will set when the counter value matches the VAL1 value. xx_xxx1b - PWM_OUT_TRIG0 will set when the counter value matches the VAL0 value.

28.5.17 Fault Disable Mapping Register 0 (SM0DISMAP0 - SM2DISMAP0)

Offset

Register	Offset
SM0DISMAP0	2Ch
SM1DISMAP0	8Ch
SM2DISMAP0	ECh

Function

This register determines which PWM pins are disabled by the fault protection inputs. Reset sets all of the bits in the fault disable mapping register. This register is write-protected by MCTRL2[WRPROT] bits.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	1				DIS0X				DIS0B				DIS0A			
W																
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Fields

Field	Function
15-12 —	Reserved
11-8 DIS0X	PWM_X Fault Disable Mask 0 Each of the four bits of this field is one-to-one associated with the four FAULTx inputs of fault channel 0. The PWM_X output is turned off if there is a logic 1 on a FAULTx input and a 1 in the corresponding bit of this field. DIS0X[0] is associated with FAULT0. DIS0X[1] is associated with FAULT1. DIS0X[2] is associated with FAULT2. DIS0X[3] is associated with FAULT3. A reset sets all bits in this field.
7-4 DIS0B	PWM_B Fault Disable Mask 0 Each of the four bits of this field is one-to-one associated with the four FAULTx inputs of fault channel 0. The PWM_B output is turned off if there is a logic 1 on a FAULTx input and a 1 in the corresponding bit of this field. DIS0B[0] is associated with FAULT0. DIS0B[1] is associated with FAULT1. DIS0B[2] is associated with FAULT2. DIS0B[3] is associated with FAULT3. A reset sets all bits in this field.
3-0 DIS0A	PWM_A Fault Disable Mask 0 Each of the four bits of this field is one-to-one associated with the four FAULTx inputs of fault channel 0. The PWM_A output is turned off if there is a logic 1 on a FAULTx input and a 1 in the corresponding bit of this field. DIS0A[0] is associated with FAULT0. DIS0A[1] is associated with FAULT1. DIS0A[2] is associated with FAULT2. DIS0A[3] is associated with FAULT3. A reset sets all bits in this field.

28.5.18 Deadtime Count Register 0 (SM0DTCNT0 - SM2DTCNT0)

Offset

Register	Offset
SM0DTCNT0	30h
SM1DTCNT0	90h
SM2DTCNT0	F0h

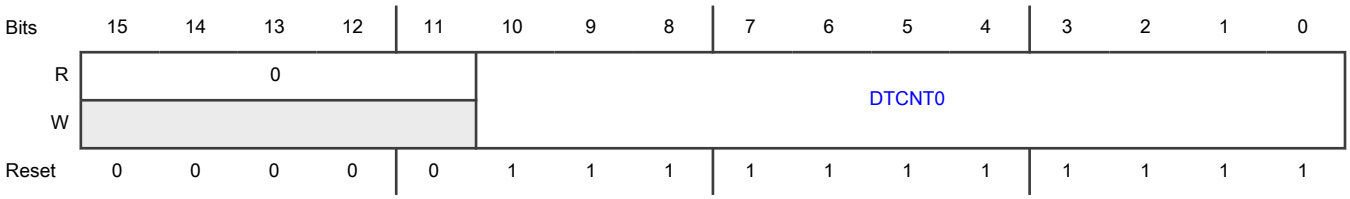
All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Function

Deadtime operation applies only to complementary channel operation. The values written to the DTCNTx registers are in terms of IPBus clock cycles regardless of the setting of CTRL[PRSC] and/or CTRL2[CLK_SEL]. Reset sets the deadtime count registers to a default value of 0x07FF, selecting a deadtime of 2047 IPBus clock cycles, when fractional delay is not enabled. The DTCNTx registers are not byte accessible.

Diagram



Fields

Field	Function
15-11 —	Reserved
10-0 DTCNT0	Deadtime Count Register 0 This field is used to control the deadtime during 0 to 1 transitions of the PWM_A output (assuming normal polarity).

28.5.19 Deadtime Count Register 1 (SM0DTCNT1 - SM2DTCNT1)

Offset

Register	Offset
SM0DTCNT1	32h
SM1DTCNT1	92h
SM2DTCNT1	F2h

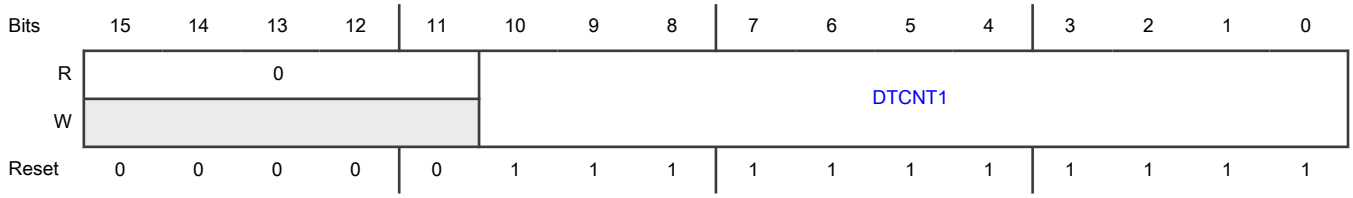
Function

Deadtime operation applies only to complementary channel operation. The values written to the DTCNTx registers are in terms of IPBus clock cycles regardless of the setting of CTRL[PRSC] and/or CTRL2[CLK_SEL]. Reset sets the deadtime count registers to a default value of 0x07FF, selecting a deadtime of 2047 IPBus clock cycles, when fractional delay is not enabled. The DTCNTx registers are not byte accessible.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram



Fields

Field	Function
15-11 —	Reserved
10-0 DTCNT1	Deadtime Count Register 1 This field is used to control the deadtime during 0 to 1 transitions of the complementary PWM_B output.

28.5.20 Capture Control X Register (SM0CAPTCTRLX)

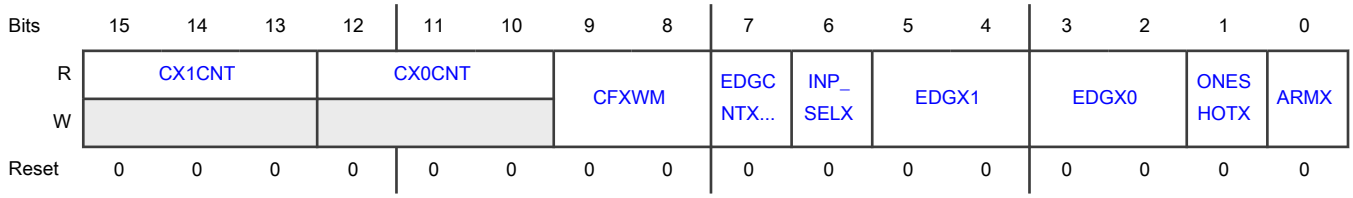
Offset

Register	Offset
SM0CAPTCTRLX	3Ch

Function

Contains capture controls for mode X. This register is write-protected by MCTRL2[WRPROT] bits.

Diagram



Fields

Field	Function
15-13 CX1CNT	Capture X1 FIFO Word Count This field reflects the number of words in the Capture X1 FIFO. (FIFO depth is 1.)
12-10 CX0CNT	Capture X0 FIFO Word Count This field reflects the number of words in the Capture X0 FIFO. (FIFO depth is 1.)

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
9-8 CFXWM	Capture X FIFOs Water Mark This field represents the water mark level for capture X FIFOs. The capture flags, STS[CFX1] and STS[CFX0], are not set until the word count of the corresponding FIFO is greater than or equal to this water mark level. (FIFO depth is 1.)
7 EDGCNTX_EN	Edge Counter X Enable This bit enables the edge counter which counts rising and falling edges on the PWM_X input signal. 0b - Edge counter disabled and held in reset 1b - Edge counter enabled
6 INP_SELX	Input Select X This bit selects between the raw PWM_X input signal and the output of the edge counter/compare circuitry as the source for the input capture circuit. 0b - Raw PWM_X input signal selected as source. 1b - Edge Counter. Output of edge counter/compare selected as source. When this bitfield is set to 1, the internal edge counter is enabled and the rising and/or falling edges specified by the CAPTCTRLX[EDGX0] and CAPTCTRLX[EDGX1] fields are ignored. The software must place a value other than 00 in either or both of the CAPTCTLRX[EDGX0] and/or CAPTCTRLX[EDGX1] fields to enable one or both of the capture registers.
5-4 EDGX1	Edge X 1 These bits control the input capture 1 circuitry by determining which input edges cause a capture event. 00b - Disabled 01b - Capture falling edges 10b - Capture rising edges 11b - Capture any edge
3-2 EDGX0	Edge X 0 These bits control the input capture 0 circuitry by determining which input edges cause a capture event. 00b - Disabled 01b - Capture falling edges 10b - Capture rising edges 11b - Capture any edge
1 ONESHOTX	One Shot Mode Aux This field selects between free running and one shot mode for the input capture circuitry. 0b - Free Running. Free running mode is selected if both capture circuits are enabled, then capture circuit 0 is armed first after the ARMX bit is set. Once a capture occurs, capture circuit 0 is disarmed, and capture circuit 1 is armed. After capture circuit 1 performs a capture, it is

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	disarmed, and capture circuit 0 is re-armed. The process continues indefinitely. If only one of the capture circuits is enabled, then captures continue indefinitely on the enabled capture circuit. 1b - One Shot. One shot mode is selected if both capture circuits are enabled, then capture circuit 0 is armed first after the ARMX bit is set. Once a capture occurs, capture circuit 0 is disarmed, and capture circuit 1 is armed. After capture circuit 1 performs a capture, it is disarmed, and the ARMX bit is cleared. No further captures are performed until the ARMX bit is set again. If only one of the capture circuits is enabled, then a single capture occurs on the enabled capture circuit and the ARMX bit is then cleared.
0 ARMX	Arm X Setting this bit high starts the input capture process. This bit can be cleared at any time to disable input capture operation. This bit is self-cleared when in one shot mode and one or more of the enabled capture circuits has had a capture event. 0b - Input capture operation is disabled. 1b - Input capture operation as specified by CAPTCTRLX[EDGXx] is enabled.

28.5.21 Capture Compare X Register (SM0CAPTCOMPX)

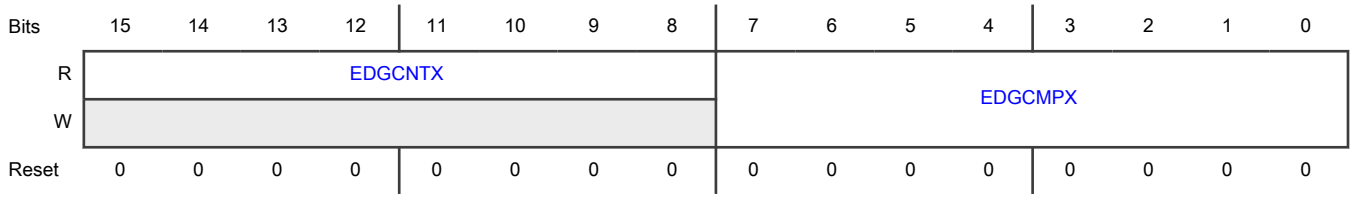
Offset

Register	Offset
SM0CAPTCOMPX	3Eh

Function

Contains capture and control values for mode X. This register is write-protected by MCTRL2[WRPROT] bits.

Diagram



Fields

Field	Function
15-8 EDGCNTX	Edge Counter X This field contains the edge counter value for the PWM_X input capture circuitry.
7-0 EDGCMPX	Edge Compare X This field is the compare value associated with the edge counter for the PWM_X input capture circuitry.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

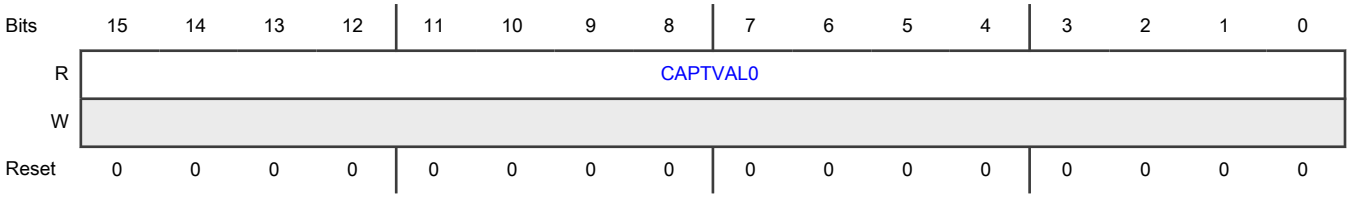
28.5.22 Capture Value 0 Register (SM0CVAL0)

Offset

Register	Offset
SM0CVAL0	40h

Function
Writing this register generates bus transfer error.

Diagram



Fields

Field	Function
15-0 CAPTVAL0	Capture Value 0 This field stores the value captured from the submodule counter. Exactly when this capture occurs is defined by CAPTCTRLX[EDGX0]. Each capture increases the value of CAPTCTRLX[CX0CNT] by 1 until the maximum value is reached. Each read of this field decreases the value of CAPTCTRLX[CX0CNT] by 1 until 0 is reached. This field is not byte accessible.

28.5.23 Capture Value 0 Cycle Register (SM0CVAL0CYC)

Offset

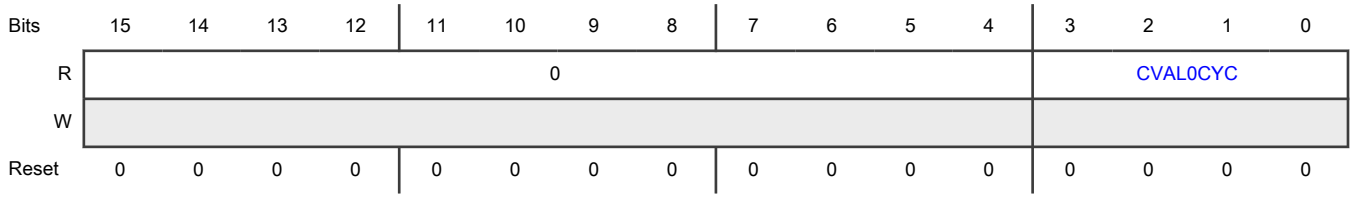
Register	Offset
SM0CVAL0CYC	42h

Function
Writing this register generates bus transfer error.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram



Fields

Field	Function
15-4 —	Reserved
3-0 CVAL0CYC	Capture Value 0 Cycle This field stores the cycle number corresponding to the value captured in CVAL0. This field is incremented each time the counter is loaded with the INIT value at the end of a PWM modulo cycle.

28.5.24 Capture Value 1 Register (SM0CVAL1)

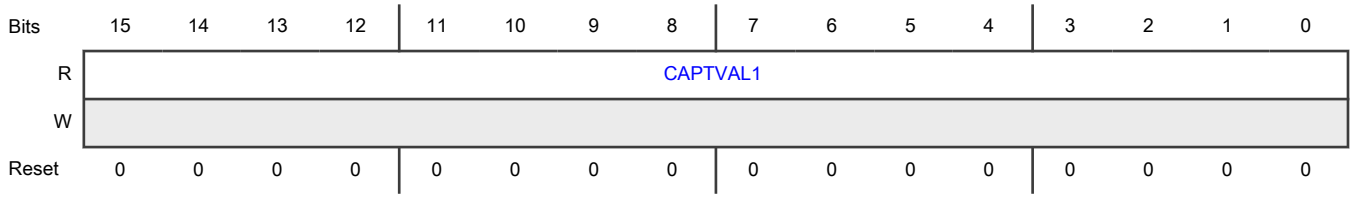
Offset

Register	Offset
SM0CVAL1	44h

Function

Writing this register generates bus transfer error.

Diagram



Fields

Field	Function
15-0 CAPTVAL1	Capture Value 1 This field stores the value captured from the submodule counter when this capture occurs is defined by CAPTCTRLX[EDGX1]. Each capture increases the value of CAPTCTRLX[CX1CNT] by 1 until the maximum value is reached. Each read of this field decreases the value of CAPTCTRLX[CX1CNT] by 1 until 0 is reached. This field is not byte accessible.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

28.5.25 Capture Value 1 Cycle Register (SM0CVAL1CYC)

Offset

Register	Offset
SM0CVAL1CYC	46h

Function

Writing this register generates bus transfer error.

Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0												CVAL1CYC			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
15-4 —	Reserved
3-0 CVAL1CYC	Capture Value 1 Cycle This field stores the cycle number corresponding to the value captured in CVAL1. This field is incremented each time the counter is loaded with the INIT value at the end of a PWM modulo cycle. Resets to 0 at POR or hard reset.

28.5.26 Capture PWM_X Input Filter Register (SM0CAPTFILTX - SM2CAPTFILTX)

Offset

Register	Offset
SM0CAPTFILTX	5Eh
SM1CAPTFILTX	BEh
SM2CAPTFILTX	11Eh

Function

Input filter considerations include:

- The CAPTX_FILT_PER value should be set such that the sampling period is larger than the period of the expected noise. This way a noise spike will only corrupt one sample. The CAPTX_FILT_CNT value should be chosen to reduce the probability of noisy samples causing an incorrect transition to be recognized. The probability of an incorrect transition is defined as the probability of an incorrect sample raised to the CAPTX_FILT_CNT+3 power.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

- The values of FILT_PER and CAPTX_FILT_CNT must also be traded off against the desire for minimal latency in recognizing input transitions. Turning on the input filter (setting CAPTX_FILT_PER to a non-zero value) introduces a latency of ((CAPTX_FILT_CNT+4) x CAPTX_FILT_PER x IPBus clock period).

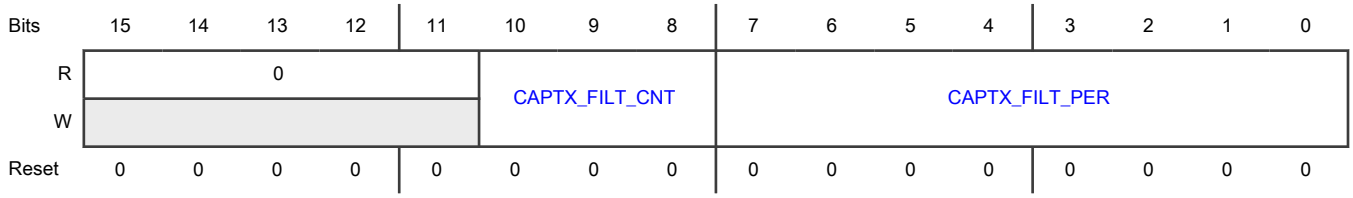
NOTE

When the filter is enabled, there is a combinational path to disable the PWM outputs to ensure rapid response to input capture conditions if the eFlexPWM module loses its clock. The latency induced by the filter will be seen in the time to set FSTS[FFLAG] and FSTS[FFPIN].

This register is not byte accessible

This register is write-protected by MCTRL2[WRPROT] bits.

Diagram



Fields

Field	Function
15-11 —	Reserved
10-8 CAPTX_FILT_CNT	Input Capture Filter Count These bits represent the number of consecutive samples that must agree prior to the input filter accepting an input transition. The number of samples is the decimal value of this field plus three: the bit field value of 0-7 represents 3-10 samples, respectively. The value of CAPTX_FILT_CNT affects the input latency.
7-0 CAPTX_FILT_PER	Input Capture Filter Period This field applies universally to all capture inputs. These bits represent the sampling period (in IPBus clock cycles) of the input capture pin input filter. Each input is sampled multiple times at the rate specified by this field. If CAPTX_FILT_PER is 0x00 (default), then the input filter is bypassed. The value of CAPTX_FILT_PER affects the input latency. **NOTE** When changing values for CAPTX_FILT_PER from one non-zero value to another non-zero value, first write a value of zero to clear the filter.

28.5.27 Capture Control X Register (SM1CAPTCTRLX)

Offset

Register	Offset
SM1CAPTCTRLX	9Ch

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Function

Contains capture controls for mode X. This register is write-protected by MCTRL2[WRPROT] bits.

Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CX1CNT			CX0CNT			CFXWM		EDGCNTX...	INP_SELX	EDGX1		EDGX0		ONESHOTX	ARMX
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
15-13 CX1CNT	Capture X1 FIFO Word Count This field reflects the number of words in the Capture X1 FIFO. (FIFO depth is 1.)
12-10 CX0CNT	Capture X0 FIFO Word Count This field reflects the number of words in the Capture X0 FIFO. (FIFO depth is 1.)
9-8 CFXWM	Capture X FIFOs Water Mark This field represents the water mark level for capture X FIFOs. The capture flags, STS[CFX1] and STS[CFX0], are not set until the word count of the corresponding FIFO is greater than or equal to this water mark level. (FIFO depth is 1.)
7 EDGCNTX_EN	Edge Counter X Enable This bit enables the edge counter which counts rising and falling edges on the PWM_X input signal. 0b - Edge counter disabled and held in reset 1b - Edge counter enabled
6 INP_SELX	Input Select X This bit selects between the raw PWM_X input signal and the output of the edge counter/compare circuitry as the source for the input capture circuit. 0b - Raw PWM_X input signal selected as source. 1b - Edge Counter. Output of edge counter/compare selected as source. When this bitfield is set to 1, the internal edge counter is enabled and the rising and/or falling edges specified by the CAPTCTRLX[EDGX0] and CAPTCTRLX[EDGX1] fields are ignored. The software must place a value other than 00 in either or both of the CAPTCTLRX[EDGX0] and/or CAPTCTRLX[EDGX1] fields to enable one or both of the capture registers.
5-4 EDGX1	Edge X 1 These bits control the input capture 1 circuitry by determining which input edges cause a capture event. 00b - Disabled 01b - Capture falling edges

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	10b - Capture rising edges 11b - Capture any edge
3-2 EDGX0	Edge X 0 These bits control the input capture 0 circuitry by determining which input edges cause a capture event. 00b - Disabled 01b - Capture falling edges 10b - Capture rising edges 11b - Capture any edge
1 ONESHOTX	One Shot Mode Aux This field selects between free running and one shot mode for the input capture circuitry. 0b - Free Running. Free running mode is selected if both capture circuits are enabled, then capture circuit 0 is armed first after the ARMX bit is set. Once a capture occurs, capture circuit 0 is disarmed, and capture circuit 1 is armed. After capture circuit 1 performs a capture, it is disarmed, and capture circuit 0 is re-armed. The process continues indefinitely. If only one of the capture circuits is enabled, then captures continue indefinitely on the enabled capture circuit. 1b - One Shot. One shot mode is selected if both capture circuits are enabled, then capture circuit 0 is armed first after the ARMX bit is set. Once a capture occurs, capture circuit 0 is disarmed, and capture circuit 1 is armed. After capture circuit 1 performs a capture, it is disarmed, and the ARMX bit is cleared. No further captures are performed until the ARMX bit is set again. If only one of the capture circuits is enabled, then a single capture occurs on the enabled capture circuit and the ARMX bit is then cleared.
0 ARMX	Arm X Setting this bit high starts the input capture process. This bit can be cleared at any time to disable input capture operation. This bit is self-cleared when in one shot mode and one or more of the enabled capture circuits has had a capture event. 0b - Input capture operation is disabled. 1b - Input capture operation as specified by CAPTCTRLX[EDGXx] is enabled.

28.5.28 Capture Compare X Register (SM1CAPTCOMPX)

Offset

Register	Offset
SM1CAPTCOMPX	9Eh

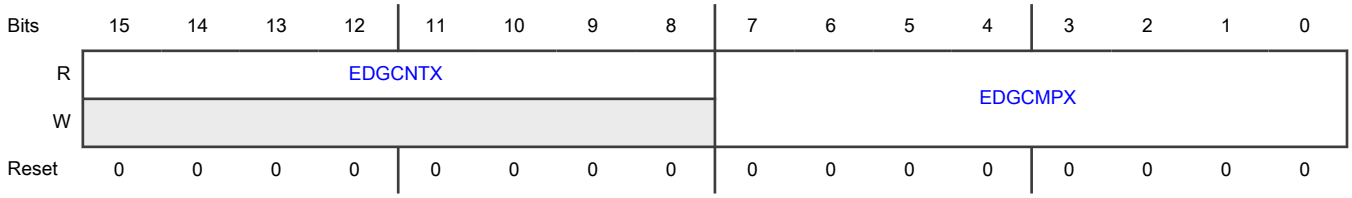
Function

Contains capture and control values for mode X. This register is write-protected by MCTRL2[WRPROT] bits.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram



Fields

Field	Function
15-8 EDGCNTX	Edge Counter X This field contains the edge counter value for the PWM_X input capture circuitry.
7-0 EDGCMPX	Edge Compare X This field is the compare value associated with the edge counter for the PWM_X input capture circuitry.

28.5.29 Capture Value 0 Register (SM1CVAL0)

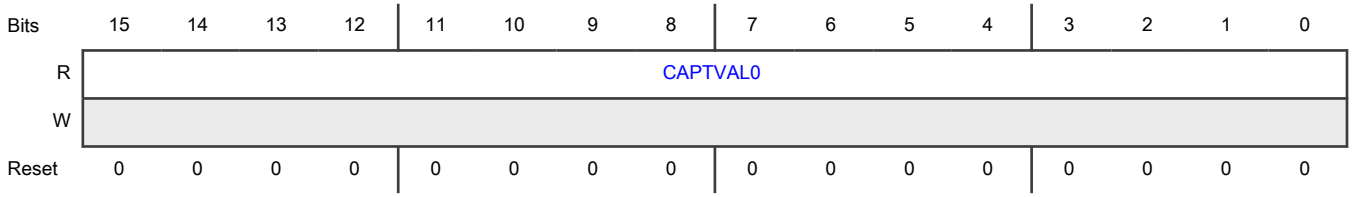
Offset

Register	Offset
SM1CVAL0	A0h

Function

Writing this register generates bus transfer error.

Diagram



Fields

Field	Function
15-0 CAPTVAL0	Capture Value 0 This field stores the value captured from the submodule counter. Exactly when this capture occurs is defined by CAPTCTRLX[EDGX0]. Each capture increases the value of CAPTCTRLX[CX0CNT] by 1 until the maximum value is reached. Each read of this field decreases the value of CAPTCTRLX[CX0CNT] by 1 until 0 is reached. This field is not byte accessible.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

28.5.30 Capture Value 0 Cycle Register (SM1CVAL0CYC)

Offset

Register	Offset
SM1CVAL0CYC	A2h

Function
Writing this register generates bus transfer error.

Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0												CVAL0CYC			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
15-4 —	Reserved
3-0 CVAL0CYC	Capture Value 0 Cycle This field stores the cycle number corresponding to the value captured in CVAL0. This field is incremented each time the counter is loaded with the INIT value at the end of a PWM modulo cycle.

28.5.31 Capture Value 1 Register (SM1CVAL1)

Offset

Register	Offset
SM1CVAL1	A4h

Function
Writing this register generates bus transfer error.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CAPTVAL1															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
15-0 CAPTVAL1	Capture Value 1 This field stores the value captured from the submodule counter when this capture occurs is defined by CAPTCTRLX[EDGX1]. Each capture increases the value of CAPTCTRLX[CX1CNT] by 1 until the maximum value is reached. Each read of this field decreases the value of CAPTCTRLX[CX1CNT] by 1 until 0 is reached. This field is not byte accessible.

28.5.32 Capture Value 1 Cycle Register (SM1CVAL1CYC)

Offset

Register	Offset
SM1CVAL1CYC	A6h

Function

Writing this register generates bus transfer error.

Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0												CVAL1CYC			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
15-4 —	Reserved
3-0 CVAL1CYC	Capture Value 1 Cycle This field stores the cycle number corresponding to the value captured in CVAL1. This field is incremented each time the counter is loaded with the INIT value at the end of a PWM modulo cycle. Resets to 0 at POR or hard reset.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

28.5.33 Phase Delay Register (SM1PHASEDLY - SM2PHASEDLY)

Offset

Register	Offset
SM1PHASEDLY	B8h
SM2PHASEDLY	118h

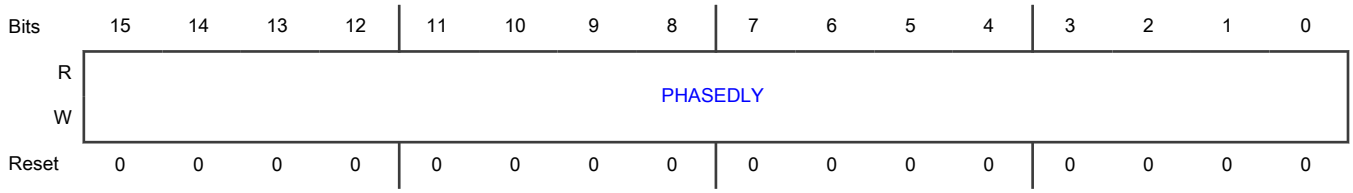
Function

The 16-bit unsigned value in this buffered register defines the delay from the master sync signal of submodule 0 to the time that this submodule recognizes the master sync in PWM clock periods. CTRL2[INIT_SEL] must be set to 10b to select the master sync signal as the source for initialization when using this register. Setting this register with a non-zero value and using the master sync signal as the initialization source, allows the output of this submodule to be a fixed number of cycles delayed from submodule 0. For PWM operation, the buffered contents of this register are updated at the start of every PWM cycle. This field is not byte accessible.

NOTE

The PHASEDLY register is buffered. The value written does not take effect until MCTRL[LDOK] is set and the next PWM load cycle begins or CTRL[LDMOD] is set. This register cannot be written when MCTRL[LDOK] is set. Reading PHASEDLY reads the value in a buffer and not necessarily the value that the PWM generator is currently using. Also, the value of this register should not be set to a value larger than the period defined in submodule 0.

Diagram



Fields

Field	Function
15-0 PHASEDLY	Initial Count Register Bits

28.5.34 Capture Control X Register (SM2CAPTCTRLX)

Offset

Register	Offset
SM2CAPTCTRLX	FCh

Function

Contains capture controls for mode X. This register is write-protected by MCTRL2[WRPROT] bits.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CX1CNT				CX0CNT				EDGCNTX...	INP_SELX	EDGX1		EDGX0		ONESHOTX	ARMX
W							CFXWM									
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
15-13 CX1CNT	Capture X1 FIFO Word Count This field reflects the number of words in the Capture X1 FIFO. (FIFO depth is 1.)
12-10 CX0CNT	Capture X0 FIFO Word Count This field reflects the number of words in the Capture X0 FIFO. (FIFO depth is 1.)
9-8 CFXWM	Capture X FIFOs Water Mark This field represents the water mark level for capture X FIFOs. The capture flags, STS[CFX1] and STS[CFX0], are not set until the word count of the corresponding FIFO is greater than or equal to this water mark level. (FIFO depth is 1.)
7 EDGCNTX_EN	Edge Counter X Enable This bit enables the edge counter which counts rising and falling edges on the PWM_X input signal. 0b - Edge counter disabled and held in reset 1b - Edge counter enabled
6 INP_SELX	Input Select X This bit selects between the raw PWM_X input signal and the output of the edge counter/compare circuitry as the source for the input capture circuit. 0b - Raw PWM_X input signal selected as source. 1b - Edge Counter. Output of edge counter/compare selected as source. When this bitfield is set to 1, the internal edge counter is enabled and the rising and/or falling edges specified by the CAPTCTRLX[EDGX0] and CAPTCTRLX[EDGX1] fields are ignored. The software must place a value other than 00 in either or both of the CAPTCTLRX[EDGX0] and/or CAPTCTRLX[EDGX1] fields to enable one or both of the capture registers.
5-4 EDGX1	Edge X 1 These bits control the input capture 1 circuitry by determining which input edges cause a capture event. 00b - Disabled 01b - Capture falling edges 10b - Capture rising edges 11b - Capture any edge
3-2	Edge X 0

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
EDGX0	These bits control the input capture 0 circuitry by determining which input edges cause a capture event. 00b - Disabled 01b - Capture falling edges 10b - Capture rising edges 11b - Capture any edge
1 ONESHOTX	One Shot Mode Aux This field selects between free running and one shot mode for the input capture circuitry. 0b - Free Running. Free running mode is selected if both capture circuits are enabled, then capture circuit 0 is armed first after the ARMX bit is set. Once a capture occurs, capture circuit 0 is disarmed, and capture circuit 1 is armed. After capture circuit 1 performs a capture, it is disarmed, and capture circuit 0 is re-armed. The process continues indefinitely. If only one of the capture circuits is enabled, then captures continue indefinitely on the enabled capture circuit. 1b - One Shot. One shot mode is selected if both capture circuits are enabled, then capture circuit 0 is armed first after the ARMX bit is set. Once a capture occurs, capture circuit 0 is disarmed, and capture circuit 1 is armed. After capture circuit 1 performs a capture, it is disarmed, and the ARMX bit is cleared. No further captures are performed until the ARMX bit is set again. If only one of the capture circuits is enabled, then a single capture occurs on the enabled capture circuit and the ARMX bit is then cleared.
0 ARMX	Arm X Setting this bit high starts the input capture process. This bit can be cleared at any time to disable input capture operation. This bit is self-cleared when in one shot mode and one or more of the enabled capture circuits has had a capture event. 0b - Input capture operation is disabled. 1b - Input capture operation as specified by CAPTCTRLX[EDGXx] is enabled.

28.5.35 Capture Compare X Register (SM2CAPTCOMPX)

Offset

Register	Offset
SM2CAPTCOMPX	FEh

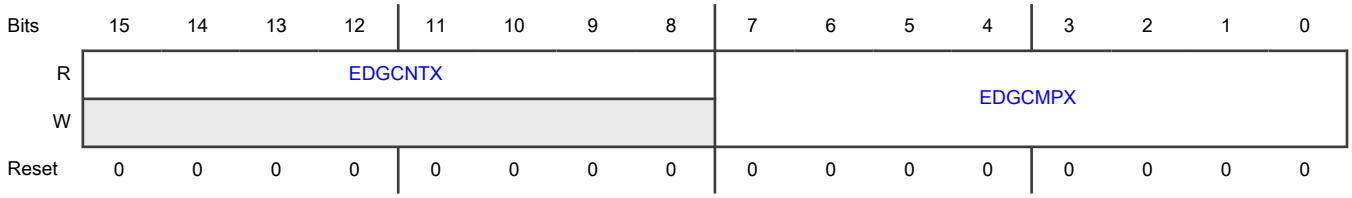
Function

Contains capture and control values for mode X. This register is write-protected by MCTRL2[WRPROT] bits.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram



Fields

Field	Function
15-8 EDGCNTX	Edge Counter X This field contains the edge counter value for the PWM_X input capture circuitry.
7-0 EDGCMPX	Edge Compare X This field is the compare value associated with the edge counter for the PWM_X input capture circuitry.

28.5.36 Capture Value 0 Register (SM2CVAL0)

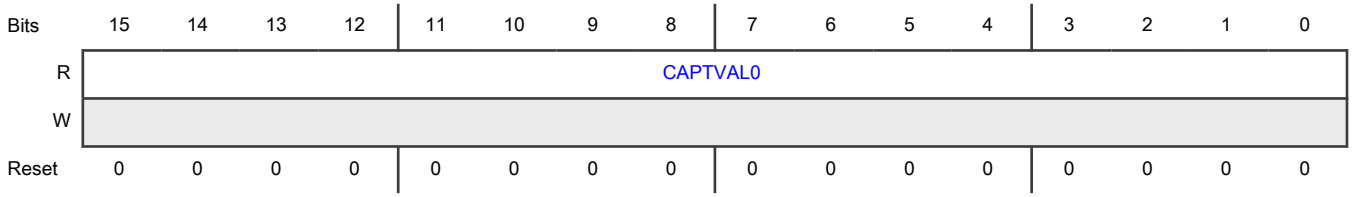
Offset

Register	Offset
SM2CVAL0	100h

Function

Writing this register generates bus transfer error.

Diagram



Fields

Field	Function
15-0 CAPTVAL0	Capture Value 0 This field stores the value captured from the submodule counter. Exactly when this capture occurs is defined by CAPTCTRLX[EDGX0]. Each capture increases the value of CAPTCTRLX[CX0CNT] by 1 until the maximum value is reached. Each read of this field decreases the value of CAPTCTRLX[CX0CNT] by 1 until 0 is reached. This field is not byte accessible.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

28.5.37 Capture Value 0 Cycle Register (SM2CVAL0CYC)

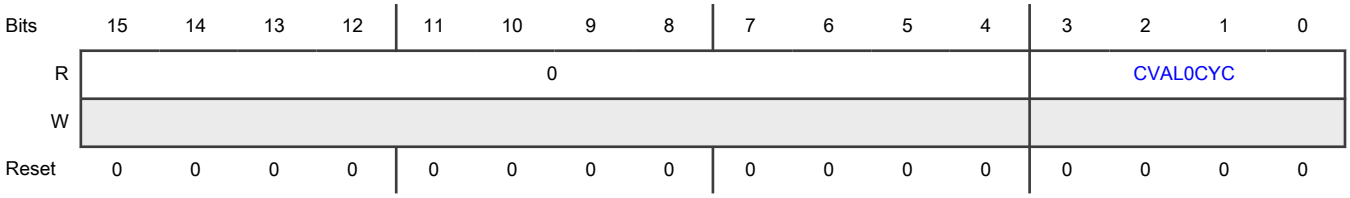
Offset

Register	Offset
SM2CVAL0CYC	102h

Function

Writing this register generates bus transfer error.

Diagram



Fields

Field	Function
15-4 —	Reserved
3-0 CVAL0CYC	Capture Value 0 Cycle This field stores the cycle number corresponding to the value captured in CVAL0. This field is incremented each time the counter is loaded with the INIT value at the end of a PWM modulo cycle.

28.5.38 Capture Value 1 Register (SM2CVAL1)

Offset

Register	Offset
SM2CVAL1	104h

Function

Writing this register generates bus transfer error.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CAPTVAL1															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
15-0 CAPTVAL1	Capture Value 1 This field stores the value captured from the submodule counter when this capture occurs is defined by CAPTCTRLX[EDGX1]. Each capture increases the value of CAPTCTRLX[CX1CNT] by 1 until the maximum value is reached. Each read of this field decreases the value of CAPTCTRLX[CX1CNT] by 1 until 0 is reached. This field is not byte accessible.

28.5.39 Capture Value 1 Cycle Register (SM2CVAL1CYC)

Offset

Register	Offset
SM2CVAL1CYC	106h

Function

Writing this register generates bus transfer error.

Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0												CVAL1CYC			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
15-4 —	Reserved
3-0 CVAL1CYC	Capture Value 1 Cycle This field stores the cycle number corresponding to the value captured in CVAL1. This field is incremented each time the counter is loaded with the INIT value at the end of a PWM modulo cycle. Resets to 0 at POR or hard reset.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

28.5.40 Output Enable Register (OUTEN)

Offset

Register	Offset
OUTEN	180h

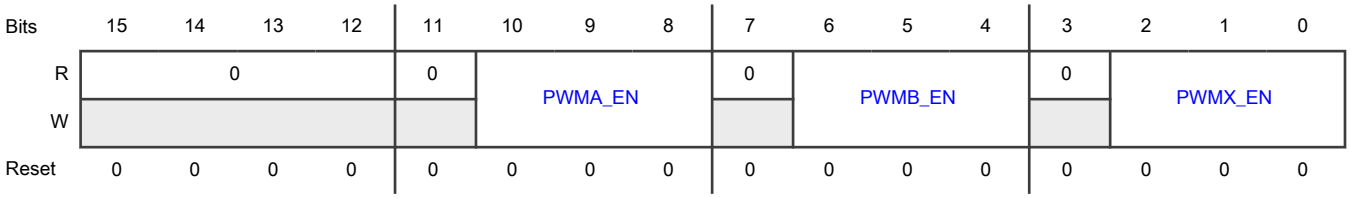
Function

Contains PWM output enables. This register is write-protected by MCTRL2[WRPROT] bits.

NOTE

It is recommended to set this register after setting other PWM configurations.

Diagram



Fields

Field	Function
15-12 —	Reserved
11 —	Reserved
10-8 PWMA_EN	PWM_A Output Enables This field enables the PWM_A outputs of submodules 2 - 0, respectively. Set these bits to 0 (output disabled) when a PWM_A pin is being used for input capture. • 0b0 PWM_A output disabled. • 0b1 PWM_A output enabled.
7 —	Reserved
6-4 PWMB_EN	PWM_B Output Enables This field enables the PWM_B outputs of submodules 2 - 0, respectively. Set these bits to 0 (output disabled) when a PWM_B pin is being used for input capture. • 0b0 PWM_B output disabled. • 0b1 PWM_B output enabled.

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
3 —	Reserved
2-0 PWMX_EN	PWM_X Output Enables This field enables the PWM_X outputs of submodules 2 - 0, respectively. Set these bits to 0 (output disabled) when a PWM_X pin is being used for input capture or deadtime correction. • 0b0 PWM_X output disabled. • 0b1 PWM_X output enabled.

28.5.41 Mask Register (MASK)

Offset

Register	Offset
MASK	182h

Function

MASK is double buffered and does not take effect until a FORCE_OUT event occurs within the appropriate submodule. Reading MASK reads the buffered values and not necessarily the values currently in effect. This double buffering can be overridden by setting the UPDATE_MASK bits.

Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0				0	MASKA			0	MASKB			0	MASKX		
W		UPDATE_MASK														
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
15 —	Reserved
14-12 UPDATE_MASK	Update Mask Bits Immediately This field masks or unmasks the PWM_X (X = A, B, or X) outputs of submodules 2 - 0 respectively by forcing the MASK* (* = A, B or X) bits to be immediately updated within submodules 2 - 0, respectively without waiting for a FORCE_OUT event. These self-clearing bits always read as zero. Software may write to any

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	or all of these bits and may set these bits in the same write operation that updates the MASKA, MASKB, and MASKX fields of this register. • 0b0 Normal operation. MASK* bits within the corresponding submodule are not updated until a FORCE_OUT event occurs within the submodule. • 0b1 Immediate operation. MASK* bits within the corresponding submodule are updated on the following clock edge after setting this bit. For example: • UPDATE_MASK[3]=1 updates the mask bits MASKA[3], MASKB[3], and MASKX[3]. • UPDATE_MASK[3:0]=0101 updates the mask bits MASKA[2], MASKA[0], MASKB[2], MASKB[0], MASKX[2], and MASKX[0].
11 —	Reserved
10-8 MASKA	PWM_A Masks This field masks the PWM_A outputs of submodules 2 - 0, respectively by forcing the output to logic 0 prior to consideration of the output polarity. • 0b0 PWM_A output normal. • 0b1 PWM_A output masked. MASKA[3:0] masks PWM_A_3 through PWM_A_0
7 —	Reserved
6-4 MASKB	PWM_B Masks This field masks the PWM_B outputs of submodules 2 - 0, respectively by forcing the output to logic 0 prior to consideration of the output polarity. • 0b0 PWM_B output normal. • 0b1 PWM_B output masked. MASKB[3:0] masks PWM_B_3 through PWM_B_0
3 —	Reserved
2-0 MASKX	PWM_X Masks This field masks the PWM_X outputs of submodules 2 - 0, respectively by forcing the output to logic 0 prior to consideration of the output polarity. • 0b0 PWM_X output normal. • 0b1 PWM_X output masked. MASKX[3:0] masks PWM_X_3 through PWM_X_0

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

28.5.42 Software Controlled Output Register (SWCOUT)

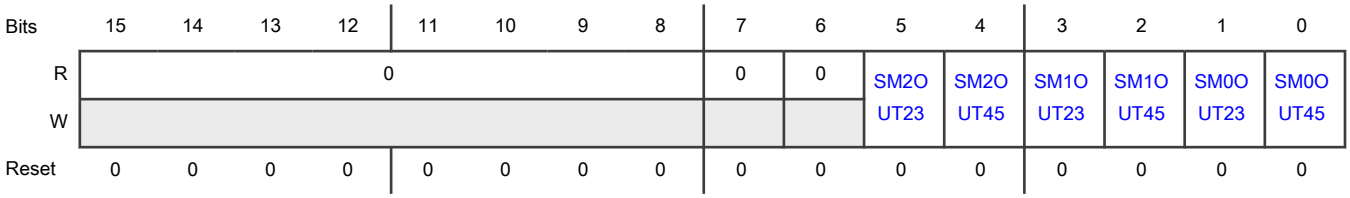
Offset

Register	Offset
SWCOUT	184h

Function

These bits are double buffered and do not take effect until a FORCE_OUT event occurs within the appropriate submodule. Reading these bits reads the buffered value and not necessarily the value currently in effect.

Diagram



Fields

Field	Function
15-8 —	Reserved
7 —	Reserved
6 —	Reserved
5 SM2OUT23	Submodule 2 Software Controlled Output 23 This field is used when DTSRCSEL[SM2SEL23] is set to b10. It allows software control of which signal is supplied to the deadtime generator of that submodule. 0b - A logic 0 is supplied to the deadtime generator of submodule 2 instead of PWM23. 1b - A logic 1 is supplied to the deadtime generator of submodule 2 instead of PWM23.
4 SM2OUT45	Submodule 2 Software Controlled Output 45 This field is used when DTSRCSEL[SM2SEL45] is set to b10. It allows software control of which signal is supplied to the deadtime generator of that submodule. 0b - A logic 0 is supplied to the deadtime generator of submodule 2 instead of PWM45. 1b - A logic 1 is supplied to the deadtime generator of submodule 2 instead of PWM45.
3	Submodule 1 Software Controlled Output 23

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
SM1OUT23	This field is used when DTSRCSEL[SM1SEL23] is set to b10. It allows software control of which signal is supplied to the deadtime generator of that submodule. 0b - A logic 0 is supplied to the deadtime generator of submodule 1 instead of PWM23. 1b - A logic 1 is supplied to the deadtime generator of submodule 1 instead of PWM23.
2 SM1OUT45	Submodule 1 Software Controlled Output 45 This field is used when DTSRCSEL[SM1SEL45] is set to b10. It allows software control of which signal is supplied to the deadtime generator of that submodule. 0b - A logic 0 is supplied to the deadtime generator of submodule 1 instead of PWM45. 1b - A logic 1 is supplied to the deadtime generator of submodule 1 instead of PWM45.
1 SM0OUT23	Submodule 0 Software Controlled Output 23 This field is used when DTSRCSEL[SM0SEL23] is set to b10. It allows software control of which signal is supplied to the deadtime generator of that submodule. 0b - A logic 0 is supplied to the deadtime generator of submodule 0 instead of PWM23. 1b - A logic 1 is supplied to the deadtime generator of submodule 0 instead of PWM23.
0 SM0OUT45	Submodule 0 Software Controlled Output 45 This field is used when DTSRCSEL[SM0SEL45] is set to b10. It allows software control of which signal is supplied to the deadtime generator of that submodule. 0b - A logic 0 is supplied to the deadtime generator of submodule 0 instead of PWM45. 1b - A logic 1 is supplied to the deadtime generator of submodule 0 instead of PWM45.

28.5.43 PWM Source Select Register (DTSRCSEL)

Offset

Register	Offset
DTSRCSEL	186h

Function

The PWM source select bits are double buffered and do not take effect until a FORCE_OUT event occurs within the appropriate submodule. Reading these bits reads the buffered value and not necessarily the value currently in effect. This register is write-protected by MCTRL2[WRPROT] bits.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		0		SM2SEL23		SM2SEL45		SM1SEL23		SM1SEL45		SM0SEL23		SM0SEL45	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
15-14 —	Reserved
13-12 —	Reserved
11-10 SM2SEL23	Submodule 2 PWM23 Control Select This field selects possible overrides to the generated SM2PWM23 signal that passes to the deadtime logic upon the occurrence of a FORCE_OUT event in that submodule. 00b - Generated SM2PWM23 signal used by the deadtime logic. 01b - Inverted generated SM2PWM23 signal used by the deadtime logic. 10b - SWCOUT[SM2OUT23] used by the deadtime logic. 11b - PWM2_EXTA signal used by the deadtime logic.
9-8 SM2SEL45	Submodule 2 PWM45 Control Select This field selects possible overrides to the generated SM2PWM45 signal that passes to the deadtime logic upon the occurrence of a FORCE_OUT event in that deadtime logic upon the occurrence of a FORCE_OUT event in that submodule. 00b - Generated SM2PWM45 signal used by the deadtime logic. 01b - Inverted generated SM2PWM45 signal used by the deadtime logic. 10b - SWCOUT[SM2OUT45] used by the deadtime logic. 11b - Reserved
7-6 SM1SEL23	Submodule 1 PWM23 Control Select This field selects possible overrides to the generated SM1PWM23 signal that passes to the deadtime logic upon the occurrence of a FORCE_OUT event in that submodule. 00b - Generated SM1PWM23 signal used by the deadtime logic. 01b - Inverted generated SM1PWM23 signal used by the deadtime logic. 10b - SWCOUT[SM1OUT23] used by the deadtime logic. 11b - PWM1_EXTA signal used by the deadtime logic.
5-4	Submodule 1 PWM45 Control Select

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
SM1SEL45	This field selects possible overrides to the generated SM1PWM45 signal that passes to the deadtime logic upon the occurrence of a FORCE_OUT event in that submodule. 00b - Generated SM1PWM45 signal used by the deadtime logic. 01b - Inverted generated SM1PWM45 signal used by the deadtime logic. 10b - SWCOUT[SM1OUT45] used by the deadtime logic. 11b - Reserved
3-2 SM0SEL23	Submodule 0 PWM23 Control Select This field selects possible overrides to the generated SM0PWM23 signal that passes to the deadtime logic upon the occurrence of a FORCE_OUT event in that submodule. 00b - Generated SM0PWM23 signal used by the deadtime logic. 01b - Inverted generated SM0PWM23 signal used by the deadtime logic. 10b - SWCOUT[SM0OUT23] used by the deadtime logic. 11b - PWM0_EXTA signal used by the deadtime logic.
1-0 SM0SEL45	Submodule 0 PWM45 Control Select This field selects possible overrides to the generated SM0PWM45 signal that passes to the deadtime logic upon the occurrence of a FORCE_OUT event in that submodule. 00b - Generated SM0PWM45 signal used by the deadtime logic. 01b - Inverted generated SM0PWM45 signal used by the deadtime logic. 10b - SWCOUT[SM0OUT45] used by the deadtime logic. 11b - Reserved

28.5.44 Master Control Register (MCTRL)

Offset

Register	Offset
MCTRL	188h

Function

In every 4-bit field in this register, each bit acts on a separate submodule. Accordingly, the description of every bit field refers to the effect of an individual bit.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	IPOL			0	RUN			0	0			0	LDOK		
W										CLDOK						
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
15 —	Reserved
14-12 IPOL	Current Polarity This field corresponds to submodules 2 - 0, respectively. Each bit selects between PWM23 and PWM45 as the source for the generation of the complementary PWM pair output for the corresponding submodule. MCTRL[IPOL] is ignored in independent mode. MCTRL[IPOL] does not take effect until a FORCE_OUT event takes place in the appropriate submodule. Reading MCTRL[IPOL] reads the buffered value and not necessarily the value currently in effect. 000b - PWM23 is used to generate complementary PWM pair in the corresponding submodule. 001b - PWM45 is used to generate complementary PWM pair in the corresponding submodule.
11 —	Reserved
10-8 RUN	Run This field enables the clocks to the PWM generator of submodules 2 - 0, respectively. The corresponding MCTRL[RUN] bit must be set for each submodule that is using its input capture functions or is using the local reload as its reload source. When this bit equals zero, the submodule counter is reset, and PWM outputs are held. A reset clears this field. 000b - PWM counter is stopped, but PWM outputs hold the current state. 001b - PWM counter is started in the corresponding submodule.
7 —	Reserved
6-4 CLDOK	Clear Load Okay This field corresponds to submodules 2 - 0, respectively. Each write-only bit is used to clear the corresponding bit of MCTRL[LDOK]. Write a 1 to CLDOK to clear the corresponding MCTRL[LDOK] bit. If a reload occurs within a submodule with the corresponding MCTRL[LDOK] bit set at the same time that MCTRL[CLDOK] is written, then the reload in that submodule will not be performed and MCTRL[LDOK] will be cleared. CLDOK bit is self-clearing and always reads as a 0.
3 —	Reserved

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
2-0 LDOK	Load Okay This field corresponds to submodules 2 - 0, respectively. Each read/set bit loads CTRL[PRSC] and the INIT, FRACVALx, and VALx registers of the corresponding submodule into a set of buffers. The buffered prescaler divisor, submodule counter modulus value, and PWM pulse width take effect at the next PWM reload if CTRL[LDMOD] is clear or if CTRL[LDMOD] is set. Set the corresponding MCTRL[LDOK] bit by reading it when it is logic zero and then writing a logic one to it. The VALx, FRACVALx, INIT, and CTRL[PRSC] registers of the corresponding submodule cannot be written while the corresponding MCTRL[LDOK] bit is set. In Master Reload Mode (CTRL2[RELOAD_SEL]=1), it is necessary to set the LDOK bit corresponding to submodule0; however, it is recommended to also set the LDOK bit of the slave submodules, to prevent unwanted writes to the registers in the slave submodules. The MCTRL[LDOK] bit is automatically cleared after the new values are loaded, or it can be manually cleared before a reload by writing a logic 1 to the appropriate MCTRL[CLDOK] bit. LDOK bits cannot be written with a zero. MCTRL[LDOK] can be set in DMA mode when the DMA indicates that it has completed the update of all CTRL[PRSC], INIT, FRACVALx, and VALx registers in the corresponding submodule. Reset clears LDOK field. 000b - Do not load new values. 001b - Load prescaler, modulus, and PWM values of the corresponding submodule.

28.5.45 Master Control 2 Register (MCTRL2)

Offset

Register	Offset
MCTRL2	18Ah

Function

Includes control for monitoring the PLL state and write protection of some configuration registers.

Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								STRETCH_CNT_PRSC		0		WRPROT		Reserved	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
15-8	Reserved

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
—	
7-6 STRETCH_CNT_PRSC	Stretch IPBus clock count prescaler for mux0_trig/mux1_trig/out0_trig/out1_trig/pwma_trig/pwmb_trig If user config eFlexPWM work in fast clk mode(use SoC level register, eFlexPWM input signal fast_clk_mode is high), then user can use these bits to stretch output signals mux0_trig/mux1_trig/out0_trig/out1_trig/pwma_trig/pwmb_trig. 00b - Stretch count is zero, no stretch. 01b - Stretch mux0_trig/mux1_trig/out0_trig/out1_trig/pwma_trig/pwmb_trig for 2 IPBus clock period. 10b - Stretch mux0_trig/mux1_trig/out0_trig/out1_trig/pwma_trig/pwmb_trig for 4 IPBus clock period. 11b - Stretch mux0_trig/mux1_trig/out0_trig/out1_trig/pwma_trig/pwmb_trig for 8 IPBus clock period.
5-4 —	Reserved
3-2 WRPROT	Write protect Enable write protection of some configuration registers of eFlexPWM. 00b - Write protection off (default). 01b - Write protection on. 10b - Write protection off and locked until chip reset. 11b - Write protection on and locked until chip reset.
1-0 —	Reserved

28.5.46 Fault Control Register (FCTRL0)

Offset

Register	Offset
FCTRL0	18Ch

Function

For every 4-bit field in this register, the bits act on the fault inputs in order. For example, FLVL bits 15-12 act on faults 3-0, respectively. This register is write-protected by MCTRL2[WRPROT] bits.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	FLVL				FAUTO				FSAFE				FIE			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
15-12 FLVL	Fault Level This field selects the active logic level of the individual fault inputs 3-0, respectively. A reset clears this field. 0000b - A logic 0 on the fault input indicates a fault condition. 0001b - A logic 1 on the fault input indicates a fault condition.
11-8 FAUTO	Automatic Fault Clearing This field selects automatic or manual clearing of faults 3-0, respectively. A reset clears this field. 0000b - Manual fault clearing. PWM outputs disabled by this fault are not enabled until FSTS[FFLAGx] is clear at the start of a half cycle or full cycle depending on the states of FSTS[FHALF] and FSTS[FFULL]. If neither FFULL nor FHALF is set, then the fault condition cannot be cleared. This is further controlled by FCTRL[FSAFE]. 0001b - Automatic fault clearing. PWM outputs disabled by this fault are enabled when FSTS[FFPINx] is clear at the start of a half cycle or full cycle depending on the states of FSTS[FHALF] and FSTS[FFULL] without regard to the state of FSTS[FFLAGx]. If neither FFULL nor FHALF is set, then the fault condition cannot be cleared.
7-4 FSAFE	Fault Safety Mode This field selects the safety mode during manual fault clearing. A reset clears this field. FSTS[FFPINx] may indicate that a fault condition still exists even though the actual fault signal at the FAULTx pin is clear due to the fault filter latency. 0000b - Normal mode. PWM outputs disabled by this fault are not enabled until FSTS[FFLAGx] is clear at the start of a half cycle or full cycle depending on the states of FSTS[FHALF] and FSTS[FFULL] without regard to the state of FSTS[FFPINx]. If neither FHALF nor FFULL is set, then the fault condition cannot be cleared. The PWM outputs disabled by this fault input will not be re-enabled until the actual FAULTx input signal de-asserts since the fault input will combinationally disable the PWM outputs (as programmed in DISMAPn). 0001b - Safe mode. PWM outputs disabled by this fault are not enabled until FSTS[FFLAGx] is clear and FSTS[FFPINx] is clear at the start of a half cycle or full cycle depending on the states of FSTS[FHALF] and FSTS[FFULL]. If neither FHLAF nor FFULL is set, then the fault condition cannot be cleared.
3-0 FIE	Fault Interrupt Enables This field enables CPU interrupt requests generated by the FAULTx pins. A reset clears this field.

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	NOTE The fault protection circuit is independent of the FIEx bit and is always active. If a fault is detected, the PWM outputs are disabled according to the disable mapping register. 0000b - FAULTx CPU interrupt requests disabled. 0001b - FAULTx CPU interrupt requests enabled.

28.5.47 Fault Status Register (FSTS0)

Offset

Register	Offset
FSTS0	18Eh

Function

Includes controls related to fault conditions.

Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	FHALF				FFPIN				FFULL				FFLAG			
W																
Reset	0	0	0	0	u	u	u	u	0	0	0	0	u	u	u	u

Fields

Field	Function
15-12 FHALF	Half Cycle Fault Recovery This field is used to control the timing for re-enabling the PWM outputs after a fault condition. These bits apply to both automatic and manual clearing of a fault condition. These register bits are write-protected by MCTRL2[WRPROT] bits. **NOTE** Both FHALF and FFULL can be set so that the fault recovery occurs at the start of a full cycle and at the start of a half cycle (as defined by VAL0). If neither FHALF nor FFULL is set, then no fault recovery is possible. 0000b - PWM outputs are not re-enabled at the start of a half cycle. 0001b - PWM outputs are re-enabled at the start of a half cycle (as defined by VAL0).
11-8	Filtered Fault Pins

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
FFPIN	These read-only bits reflect the current state of the filtered FAULTx pins converted to high polarity. A logic 1 indicates that a fault condition exists on the filtered FAULTx pin. A reset has no effect on this field. After the system reset de-asserts, these are the possible values of FFPIN: If FCTRL[FLVL] = 0 and FAULTx = 0, then FFPIN is set to 1. If FCTRL[FLVL] = 0 and FAULTx = 1, then FFPIN is kept as 0. If FCTRL[FLVL] = 1 and FAULTx = 0, then FFPIN is kept as 0. If FCTRL[FLVL] = 1 and FAULTx = 1, then FFPIN is set to 1.
7-4 FFULL	Full Cycle This field is used to control the timing for re-enabling the PWM outputs after a fault condition. These bits apply to both automatic and manual clearing of a fault condition. These register bits are write-protected by MCTRL2[WRPROT] bits. **NOTE** Both FHALF and FFULL can be set so that the fault recovery occurs at the start of a full cycle and at the start of a half cycle (as defined by VAL0). If neither FHALF nor FFULL is set, then no fault recovery is possible. 0000b - PWM outputs are not re-enabled at the start of a full cycle 0001b - PWM outputs are re-enabled at the start of a full cycle
3-0 FFLAG	Fault Flags These read-only flags are set within two CPU cycles after a transition to active on the FAULTx pin. Clear this bit by writing a logic one to it. A reset clears this field. While the reset value is 0, these bits may be set to 1 by the time they can be read depending on the state of the fault input signals. After the system reset de-asserts, these are the possible values of FFLAG: If FCTRL[FLVL] = 0 and FAULTx = 0, then FFLAG is set to 1. If FCTRL[FLVL] = 0 and FAULTx = 1, then FFLAG is kept as 0. If FCTRL[FLVL] = 1 and FAULTx = 0, then FFLAG is kept as 0. If FCTRL[FLVL] = 1 and FAULTx = 1, then FFLAG is set to 1. 0000b - No fault on the FAULTx pin. 0001b - Fault on the FAULTx pin.

28.5.48 Fault Filter Register (FFILT0)

Offset

Register	Offset
FFILT0	190h

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Function

The settings in this register are shared among each of the fault input filters within the fault channel.

Input filter considerations include:

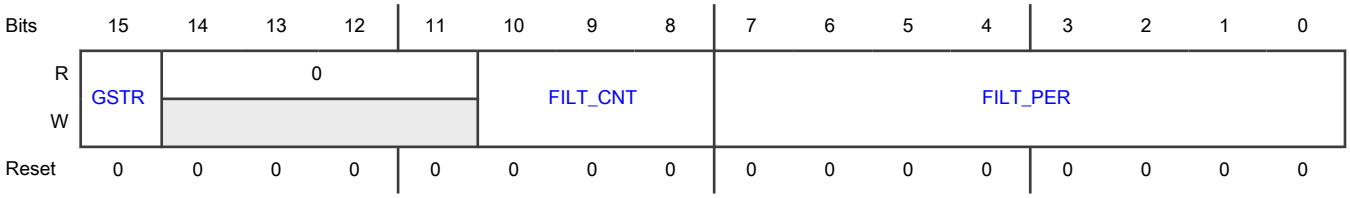
- Set the FILT_PER value such that the sampling period is larger than the period of the expected noise. This way a noise spike corrupts one sample. Select the FILT_CNT value to reduce and recognize the probability of noisy samples causing an incorrect transition. The probability of an incorrect transition is defined as the probability of an incorrect sample raised to the FILT_CNT+3 power.
- The values of FILT_PER and FILT_CNT must be traded off against the desire for minimal latency in recognizing input transitions. Turning on the input filter (setting FILT_PER to a non-zero value) introduces a latency of ((FILT_CNT+4) x FILT_PER x IPBus clock period).

NOTE

When the filter is enabled, there is a combinational path to disable the PWM outputs to ensure rapid response to fault conditions, and fault response if the eFlexPWM module loses its clock. The latency induced by the filter will be seen in the time to set FSTS[FFLAG] and FSTS[FFPIN].

This register is write-protected by MCTRL2[WRPROT] bits.

Diagram



Fields

Field	Function
15 GSTR	Fault Glitch Stretch Enable This field is used to enable the fault glitch-stretching logic. This logic ensures that narrow fault glitches are stretched to be at least 2 IPBus clock cycles wide. In some cases, a narrow fault input can cause problems due to the short PWM output shutdown/re-activation time. The stretching logic ensures that a glitch on the fault input, when the fault filter is disabled, will be registered in the fault flags. 0b - Fault input glitch stretching is disabled. 1b - Input fault signals are stretched to at least 2 IPBus clock cycles.
14-11 —	Reserved
10-8 FILT_CNT	Fault Filter Count These bits represent the number of consecutive samples that must agree prior to the input filter accepting an input transition. The number of samples is the decimal value of this field plus three: the bit field value of 0-7 represents 3-10 samples, respectively. The value of FILT_CNT affects the input latency.
7-0 FILT_PER	Fault Filter Period This 8-bit field applies universally to all fault inputs.

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	These bits represent the sampling period (in IPBus clock cycles) of the fault pin input filter. Each input is sampled multiple times at the rate specified by this field. If FILT_PER is 0x00 (default), then the input filter is bypassed. The value of FILT_PER affects the input latency. **NOTE** When changing values for FILT_PER from one non-zero value to another non-zero value, first write a value of zero to clear the filter.

28.5.49 Fault Test Register (FTST0)

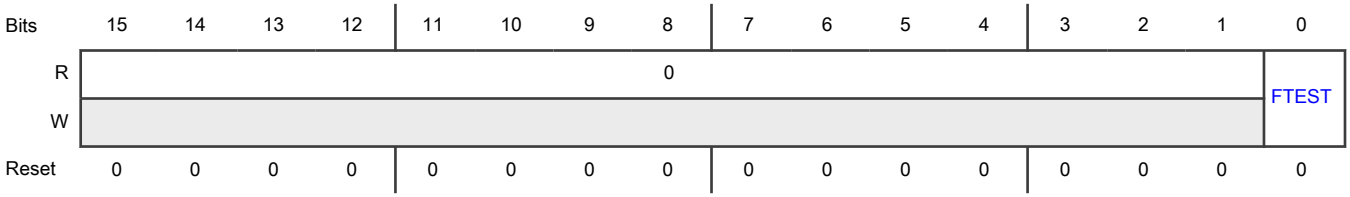
Offset

Register	Offset
FTST0	192h

Function

Contains FTEST field for fault simulation.

Diagram



Fields

Field	Function
15-1 —	Reserved
0 FTEST	Fault Test This field is used to simulate a fault condition. Setting this bit causes a simulated fault to be sent into all of the fault filters. The condition propagates to the fault flags and possibly the PWM outputs depending on the DISMAPn settings. Clearing this bit removes the simulated fault condition. This register bit is write-protected by MCTRL2[WRPROT] bits. 0b - No fault 1b - Cause a simulated fault

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

28.5.50 Fault Control 2 Register (FCTRL20)

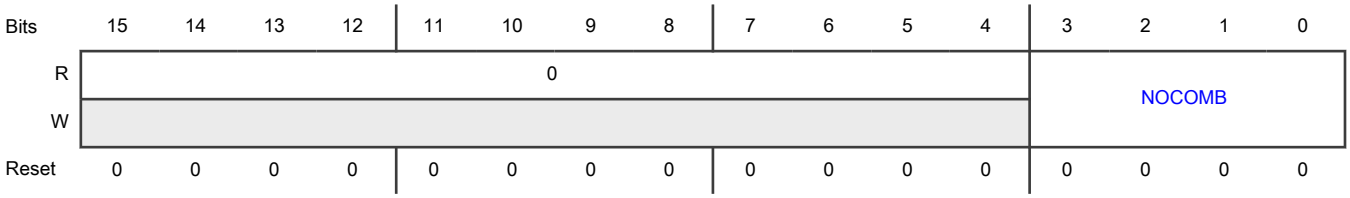
Offset

Register	Offset
FCTRL20	194h

Function

Controls combinational link from fault inputs to PWM outputs. This register is write-protected by MCTRL2[WRPROT] bits.

Diagram



Fields

Field	Function
15-4 —	Reserved
3-0 NOCOMB	No Combinational Path From Fault Input To PWM Output This field is used to control the combinational path from the fault inputs to the PWM outputs. When these bits are low (default), the corresponding fault inputs have a combinational path to the PWM outputs that are sensitive to these fault inputs (as defined by DISMAP0). This combinational path is a safety feature that ensures the output is disabled even if the SOC has a failure of its clocking system. The combinational path also means that a pulse on the fault input can cause a brief disable of the PWM output even if the fault pulse is not wide enough to get through the input filter and be latched in the fault logic. Setting these bits removes the combinational path and uses the filtered and latched fault signals as the fault source to disable the PWM outputs. This eliminates fault glitches from creating PWM output glitches but also increases the latency to respond to a real fault. 0000b - There is a combinational link from the fault inputs to the PWM outputs. The fault inputs are combined with the filtered and latched fault signals to disable the PWM outputs. 0001b - The direct combinational path from the fault inputs to the PWM outputs is disabled and the filtered and latched fault signals are used to disable the PWM outputs.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Chapter 29
Frequency Measurement (FREQME)

29.1 Chip-specific FREQME information

Table 163. Reference links to related information

Topic	Related module	Reference
Full description	FREQME	Frequency Measurement
System memory map		Memory map
Clocking		Clock distribution
Power Management		Power Management
Signal multiplexing	Port control	Signal multiplexing
Peripheral Input Mux	INPUTMUX	Peripheral Input multiplexing

29.1.1 Module instances

This chip has one instance of the FREQME module.

29.1.2 Input clock sources

Following are the input clocks for frequency measurement circuit:

- 0 Reserved
- 1 clk_in
- 2 FRO12M
- 3 fro_hf_div
- 4 Reserved
- 5 clk_16k[1]
- 6 SLOW_CLK
- 7 FREQME_CLK_IN0
- 8 FREQME_CLK_IN1
- 9 AOI0_OUT0
- 10 AOI0_OUT1
- 11 PWM0_SM0_MUX_TRIG0
- 12 PWM0_SM0_MUX_TRIG1
- 13 PWM0_SM1_MUX_TRIG0
- 14 PWM0_SM1_MUX_TRIG1
- 15 PWM0_SM2_MUX_TRIG0
- 16 PWM0_SM2_MUX_TRIG1
- 17 PWM0_SM3_MUX_TRIG0
- 18 PWM0_SM3_MUX_TRIG1

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

- 32 AOI1_OUT0
- 33 AOI1_OUT1
- 34 PWM1_SM0_MUX_TRIG0
- 35 PWM1_SM0_MUX_TRIG1
- 36 PWM1_SM1_MUX_TRIG0
- 37 PWM1_SM1_MUX_TRIG1
- 38 PWM1_SM2_MUX_TRIG0
- 39 PWM1_SM2_MUX_TRIG1
- 40 PWM1_SM3_MUX_TRIG0
- 41 PWM1_SM3_MUX_TRIG1

For details, see the FREQMEAS_REF and FREQMEAS_TAR registers in Input multiplexing (INPUTMUX) chapter.

29.2 Overview

FREQME accurately measures the frequency of an on- or off-chip target clock signal using a selectable on-chip reference clock. For example, it can accurately determine the frequency of a low-power oscillator that varies depending on process and temperature.

29.2.1 Block diagram



Figure 129. Block diagram

See the chip-specific FREQME information for the target and reference clock selection options.

29.2.2 Features

- High-accuracy Frequency Measurement mode for on- and off-chip clocks
- Pulse Width Measurement (PWM) mode
- Reference and target clock inputs selectable from among various chip-specific options
- Optional measurement complete interrupt
- Result out-of-range detection with optional interrupt

29.3 Functional description

The following sections describe FREQME functional details.

29.3.1 Frequency Measurement mode

In Frequency Measurement mode (CTL_W[PULSE_MODE] = 0), FREQME counts the number of target clock cycles that occur during a specified number of cycles from a reference clock that has a known frequency. You can then calculate the target clock frequency based on the frequency of the reference clock, the number of reference clock cycles, and the number of target clock cycles (see Equation 2).

The frequency measurement circuit is based on two 31-bit counters—one clocked by the selected reference clock and one by the selected target clock. You can only read the target clock counter. FREQME synchronizes the clocks at the start and end of

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

each count sequence during preparation for the next count sequence. A count sequence consists of incrementing the target clock counter for each target clock pulse that occurs during the time defined by $2^{CTRL\_W[REF\_SCALE]}$ periods of the reference clock.

After selecting reference and target clocks, you initiate a measurement cycle by writing 1 to CTRL_W[MEASURE_IN_PROGRESS]. You can then poll this same field, which automatically transitions to 0 when the measurement operation completes. Alternatively, the completion of the measurement operation can generate an interrupt.

The measurement cycle terminates when the reference counter equals the value $2^{CTRLSTAT[REFSCALE]}$. If CTRLSTAT[REFSCALE] = 0, the reference counter counts to one and stops. You can use this feature to measure the frequency of a fast target clock using a slow reference clock such as the 32 kHz clock without taking much time for the measurement to complete. The penalty is reduced accuracy in the measurement.

When the counting operation completes, the state of the target counter is loaded into CTRL_R[RESULT], and the CTRL_W[MEASURE_IN_PROGRESS] field is 0. You can then read the value and calculate the target frequency as follows, with the frequencies given in MHz.

$$F_{target} = (CTRL\_R[RESULT] + 1) \times F_{reference} \div 2^{CTRLSTAT[REFSCALE]}$$

Equation 2. Calculating the target clock frequency

29.3.1.1 Accuracy

The Frequency Measurement mode can measure the frequency of any on-chip (or off-chip) clock (referred to as the target clock) with a high degree of accuracy by using an on-chip clock of known frequency as a reference clock.

Uncertainty in the reference clock (for example a ±1 % accuracy of the clock) adds to the measurement error of the target clock. In general, though, this additional error is less than the uncertainty of the reference clock.

29.3.2 Pulse Measurement mode

When you write 1 to CTRL_W[PULSE_MODE] and write 1 to CTRL_W[MEASURE_IN_PROGRESS] to start a measurement cycle, FREQME counts target clock pulses while the reference clock is in a specific state (high or low), selected by writing to CTRL_W[PULSE_POL]. See the description of these fields for more details.

29.3.3 Clocking

The clock inputs are as follows:

- Reference clock (REF_CLK_IN). This clock has a known frequency.
- Target clock (TARGET_CLK_IN). This clock frequency can be calculated based on the reference clock.
- Peripheral clock used for register access (PCLK). This clock is used to write and read memory-mapped registers of this module.

NOTE

These clocks can be all synchronous.

See the chip-specific FREQME information for the target and reference clock selection options.

29.3.4 Interrupts

The following interrupts can optionally be generated at the completion of a frequency measurement cycle.

Table 164. Interrupts

Condition	Interrupt enable field	Interrupt status field
Result is ready to read	CTRL_W[RESULT_READY_INT_EN]	CTRLSTAT[RESULT_READY_INT_EN]

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table 164. Interrupts (continued)

Condition	Interrupt enable field	Interrupt status field
Result is greater than MAX[MAX_VALUE]	CTRL_W[GT_MAX_INT_EN]	CTRLSTAT[GT_MAX_INT_EN]
Result is less than MIN[MIN_VALUE]	CTRL_W[LT_MIN_INT_EN]	CTRLSTAT[LT_MIN_INT_EN]

29.4 External signals

Table 165. External signals

Signal	Description	Direction
REF_CLK_IN	Reference clock. On-chip clock with known frequency chosen as reference.	Input
TARGET_CLK_IN	Target clock. On- or off-chip clock to be measured by FREQME.	Input

29.5 Initialization

Initialize FREQME by performing the following steps:

1. Enable the clock that drives FREQME in the chip-level clock control register. This step enables the register interface and the peripheral function clock.
2. Clear the FREQME peripheral reset in the chip-level reset control register.
3. If measuring an external clock, use the Input/Output pin configuration registers to connect the FREQME target clock input (TARGET_CLK_IN) to an external pin.
4. Ensure that the reference and target clocks are enabled.
5. Select the reference and target clocks.

29.6 Memory map and register definition

This section includes the FREQME memory map and detailed descriptions of all registers.

29.6.1 FREQME register descriptions

29.6.1.1 FREQME memory map

FREQME0 base address: 4000_9000h

Offset	Register	Width (In bits)	Access	Reset value
0h	Control (in Read mode) (CTRL_R)	32	R	0000_0000h
0h	Control (in Write mode) (CTRL_W)	32	W	0000_0000h
4h	Control Status (CTRLSTAT)	32	RW	0000_0000h
8h	Minimum (MIN)	32	RW	0000_0000h
Ch	Maximum (MAX)	32	RW	7FFF_FFFFh

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

29.6.1.2 Control (in Read mode) (CTRL_R)

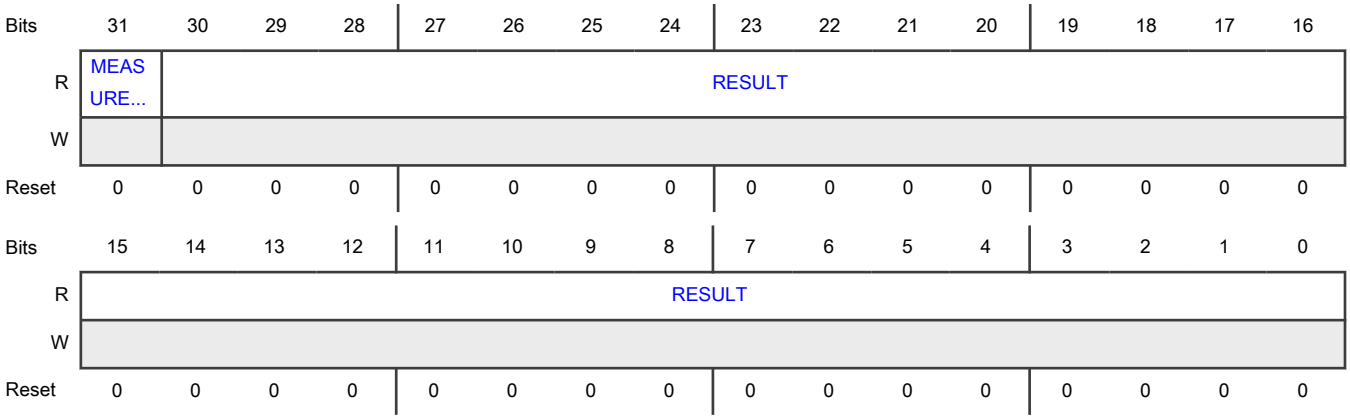
Offset

Register	Offset
CTRL_R	0h

Function

Contains different fields depending on whether you are performing a read or a write. Reading this register provides the measurement results and the status of the measurement cycle.

Diagram



Fields

Field	Function
31 MEASURE_IN_PROGRESS	Measurement In Progress Indicates whether measurement is in progress or complete. If complete, you can read the result from RESULT. 0b - Complete 1b - In progress
30-0 RESULT	Indicates the measurement result—either the target clock counter value (for Frequency Measurement mode) or pulse width measurement (for Pulse Width Measurement mode). This field is valid only when MEASURE_IN_PROGRESS = 0, In Continuous mode (CTRL_W[CONTINUOUS_MODE_EN] = 1), the value is the result from the most-recently completed measurement.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

29.6.1.3 Control (in Write mode) (CTRL_W)

Offset

Register	Offset
CTRL_W	0h

Function

Writes to this register control and configure the measurement cycle.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R																
W	MEASURE...	CONTINU...	0													
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R																
W	0	RESULT_...	GT_MAX_...	LT_MIN_...	0		PULSE_P...	PULSE_M...	0				REF_SCALE			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31 MEASURE_IN_PROGRESS	Measurement In Progress Initiates a frequency measurement cycle or terminates a measurement cycle that is in progress. Writing 1 to this field initiates a frequency or pulse width measurement process. Hardware automatically writes 0 to the MEASURE_IN_PROGRESS field when the measurement cycle completes. If there is an active measurement in progress, a new measurement starts. Writing 0 to this field forces the termination of any measurement cycle currently in progress and resets CTRL_R[RESULT] or just resets CTRL_R[RESULT] if idle. 0b - Terminates measurement 1b - Initiates measurement
30 CONTINUOUS_MODE_EN	Continuous Mode Enable When you write 1 to both MEASURE_IN_PROGRESS and this field, measurement is performed continuously. The result for the most-recently completed measurement is available in CTRL_R[RESULT]. When the result is out of range (that is, CTRLSTAT[GT_MAX_STAT] = 1 or CTRLSTAT[LT_MIN_STAT] = 1), this field automatically transitions to 0 and the continuous measurement is suspended. After clearing both status bits, you must write 1 to both MEASURE_IN_PROGRESS and this field restart measurement in continuous mode.

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	0b - Disable 1b - Enable
29-15 —	Reserved
14 RESULT_READY_INT_EN	Result Ready Interrupt Enable Generates interrupt when a measurement completes and the result is ready (CTRLSTAT[RESULT_READY_STAT] = 1). 0b - Disable 1b - Enable
13 GT_MAX_INT_EN	Greater Than Maximum Interrupt Enable Generates an interrupt when the result is greater than MAX[MAX_VALUE] (CTRLSTAT[GT_MAX_STAT] = 1). 0b - Disable 1b - Enable
12 LT_MIN_INT_EN	Less Than Minimum Interrupt Enable Generates an interrupt when the result is less than MIN[MIN_VALUE] (CTRLSTAT[LT_MIN_STAT] = 1). 0b - Disable 1b - Enable
11-10 —	Reserved
9 PULSE_POL	Pulse Polarity Specifies whether a pulse width (high period or low period) of the reference clock is measured in the Pulse Width Measurement mode. **NOTE** PULSE_POL is valid only in the Pulse Width Measurement mode. • A high period measurement is triggered by the rising edge on the reference clock input. • A low period measurement is triggered by the falling edge on the reference clock input. 0b - High period 1b - Low period
8 PULSE_MODE	Pulse Width Measurement Mode Select Selects the measurement mode—either Frequency Measurement mode or Pulse Width Measurement mode.

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	In Frequency Measurement mode, measurement begins at the rising edge of the selected reference clock and continues until a count of 2^{REF_SCALE} reference clock pulses is reached. In Pulse Width Measurement mode, the counter starts incrementing when the selected trigger edge (rising edge for high period measurement or falling edge for low period) occurs, and stops at the next edge (falling edge for a high period measurement or rising edge for a low period measurement). Select high or low period measurement by writing the desired value to PULSE_POL. 0b - Frequency Measurement mode 1b - Pulse Width Measurement mode
7-5 —	Reserved
4-0 REF_SCALE	Reference Clock Scaling Factor Specifies the reference clock scaling factor in Frequency Measurement mode. The reference count cycle is 2^{REF_SCALE}. A higher number provides better accuracy but consumes more processing time. This field is valid only in Frequency Measurement mode.

29.6.1.4 Control Status (CTRLSTAT)

Offset

Register	Offset
CTRLSTAT	4h

Function

Contains the current CTRL_W register configuration fields.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	MEASURE...	CONTINU...	0			RESULT_...	GT_MAX_...	LT_MIN_...	0							
W						W1C	W1C	W1C								
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	RESULT_...	GT_MAX_...	LT_MIN_...	0		PULSE_P...	PULSE_M...	0			REF_SCALE				
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Fields

Field	Function
31 MEASURE_IN_PROGRESS	Measurement in Progress Status Indicates the current CTRL_W[MEASURE_IN_PROGRESS] value. 0b - Not in progress 1b - In progress
30 CONTINUOUS_MODE_EN	Continuous Mode Enable Status Indicates the current CTRL_W[CONTINUOUS_MODE_EN] value. 0b - Disabled 1b - Enabled
29-27 —	Reserved
26 RESULT_READY_STAT	Result Ready Status Indicates that a measurement is complete and CTRL_R[RESULT] is ready to read. Write 1 to this field to clear. 0b - Not complete 1b - Complete
25 GT_MAX_STAT	Greater Than Maximum Result Status Indicates that a measurement is complete and the result is greater than the maximum expected value (CTRL_R[RESULT] > MAX[MAX_VALUE]). Write 1 to this field to clear. 0b - Less than MAX[MAX_VALUE] 1b - Greater than MAX[MAX_VALUE]
24 LT_MIN_STAT	Less Than Minimum Results Status Indicates that a measurement is complete and the result is less than the minimum expected value (CTRL_R[RESULT] < MIN[MIN_VALUE]). Write 1 to this field to clear. 0b - Greater than MIN[MIN_VALUE] 1b - Less than MIN[MIN_VALUE]
23-15 —	Reserved
14 RESULT_READY_INT_EN	Result Ready Interrupt Enable Indicates the current CTRL_W[RESULT_READY_INT_EN] value. 0b - Disabled 1b - Enabled
13	Greater Than Maximum Interrupt Enable

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
GT_MAX_INT_EN	Indicates the current CTRL_W[GT_MAX_INT_EN] value. 0b - Disabled 1b - Enabled
12 LT_MIN_INT_EN	Less Than Minimum Interrupt Enable Indicates the current CTRL_W[LT_MIN_INT_EN] value. 0b - Disabled 1b - Enabled
11-10 —	Reserved
9 PULSE_POL	Pulse Polarity Indicates the current CTRL_W[PULSE_POL] value. 0b - High period 1b - Low period
8 PULSE_MODE	Pulse Mode Indicates the current CTRL_W[PULSE_MODE] value. 0b - Frequency Measurement mode 1b - Pulse Width Measurement mode
7-5 —	Reserved
4-0 REF_SCALE	Reference Scale Indicates the current CTRL_W[REF_SCALE] value.

29.6.1.5 Minimum (MIN)

Offset

Register	Offset
MIN	8h

Function

Specifies the minimum expected value for the measurement result.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	MIN_VALUE														
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	MIN_VALUE															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31 —	Reserved
30-0 MIN_VALUE	Minimum Value Minimum expected value for the measurement result.

29.6.1.6 Maximum (MAX)

Offset

Register	Offset
MAX	Ch

Function

Specifies the maximum expected value for the measurement result.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	MAX_VALUE														
W																
Reset	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	MAX_VALUE															
W																
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Fields

Field	Function
31 —	Reserved
30-0 MAX_VALUE	Maximum Value Maximum expected value for the measurement result.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Chapter 30
Low-power Timer (LPTMR)

30.1 Chip-specific LPTMR information

Table 166. Reference links to related information

Topic	Related module	Reference
Full description	LPTMR	LPTMR
System memory map		Memory map
Clocking		Clock distribution
Power Management		Power Management
Signal multiplexing	Port control	Signal multiplexing

30.1.1 Module instances

This chip has one instance of the LPTMR module, LPTMR0.

30.1.2 LPTMR clocks

The prescaler and glitch filter of the LPTMR module (the module functional clock) can be clocked from one of the sources determined by LPTMRx_PSR[PCS], see Figure 68. The following table shows the clock assignments for this field.

NOTE

The chosen clock must remain enabled if the LPTMR is to continue operating in all required low-power modes.

The LPTMR allows the maximum clock frequency of 25 MHz.

Table 167. LPTMRn prescaler/glitch filter clocking options

LPTMRn_PSR[PCS]	Prescaler/glitch filter clock number	Chip clock
00	0	Reserved.
01	1	clk_16
10	2	OSC_32K
11	3	Combination of clocks configured in MRCC_LPTMR0_CLKSEL[MUX] and MRCC_LPTMR0_CLKDIV[DIV] field in SYSCON module. The Clock frequency must be less than 25 MHz to be used as a clock for the LPTMR.

30.1.3 LPTMR pulse counter input options

LPTMRn_CSR[TPS] configures the input source used in pulse counter mode. The following table shows the input assignments for this field.

All information provided in this document is subject to legal disclaimers.
© 2026 NXP B.V. All rights reserved.

Table 168. LPTMRn (n=0) pulse counter input options

LPTMRn_CSR[TPS]	Pulse counter input number	Chip input
00	0	CMP0_OUT
01	1	Reserved
10	2	LPTMRn_Alt2
11	3	LPTMRn_Alt3

30.1.4 LPTMR Reset

LPTMR is reset only on POR.

30.2 Overview

You can configure LPTMR to operate as a time counter with an optional prescaler, or as a pulse counter with an optional glitch filter, across all power modes, including low-power modes. It is reset only on POR only or Cold Reset which would include LVD/HVD sources., allowing it to be used as a time-of-day counter.

30.2.1 Block diagram

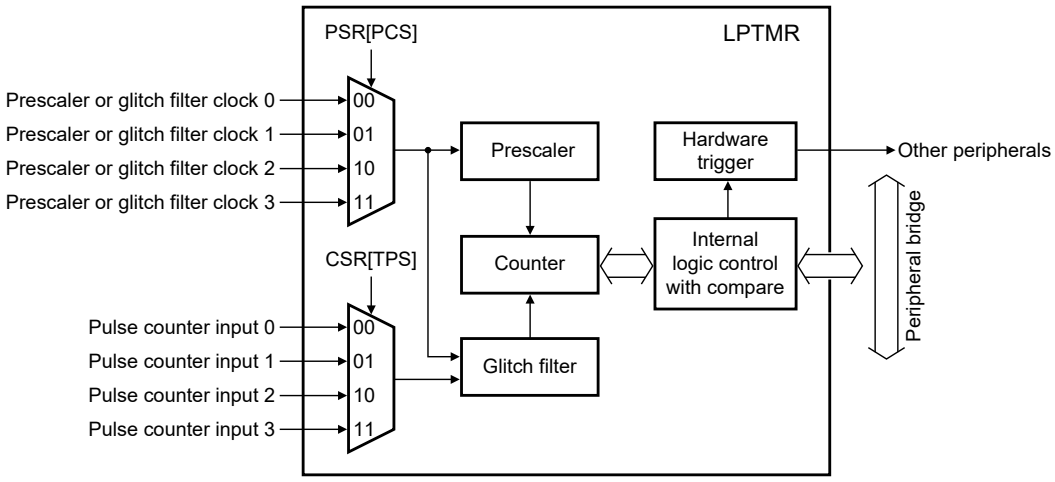


Figure 130. Block diagram

30.2.2 Features

- 32-bit time counter or pulse counter with compare:
  - Optional interrupt that can generate an asynchronous wake-up from any low-power mode
  - Hardware trigger output
  - Counter that supports a free-running mode or reset on compare
- Configurable clock source for prescaler and glitch filter
- Configurable input source for pulse counter (rising-edge or falling-edge)

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

30.3 Functional description

30.3.1 Low-power modes

In low-power modes, LPTMR continues to operate normally. You can configure LPTMR to exit a low-power mode by generating either an interrupt or a DMA request.

30.3.2 Clocks

The LPTMR prescaler and glitch filter can be clocked by one of the clocks that you configure by using PSR[PCS]. You must enable the clock source before you enable LPTMR.

In Pulse Counter mode, with the glitch filter bypassed, the selected input source directly clocks Counter (CNR), and no other clock source is required. To minimize power in this case, configure the glitch filter clock source for a clock that is disabled.

NOTE

- You may need to configure the clock source that you select in PSR[PCS] for it to remain enabled in low-power modes. Otherwise, LPTMR does not operate in low-power modes.
- The clock source or pulse input source selected for LPTMR must not exceed the maximum frequency of f_{LPTMR} defined in the chip data sheet.

30.3.3 Reset

LPTMR is reset only on POR only or Cold Reset which would include LVD/HVD sources.. When configuring LPTMR registers, you must initially write to Control Status (CSR) with LPTMR disabled, before configuring Prescaler and Glitch Filter (PSR) and Compare (CMR). Then, you must write 1 to CSR[TEN] as the last step in the initialization. Doing so ensures that LPTMR is configured correctly and the LPTMR counter is reset to zero following a POR only or Cold Reset which would include LVD/HVD sources..

30.3.4 Prescaler and glitch filter

The LPTMR prescaler and glitch filter share the same logic, which operates as a prescaler in Time Counter mode and as a glitch filter in Pulse Counter mode.

NOTE

You must not alter the prescaler and glitch filter configuration when LPTMR is enabled.

30.3.4.1 Prescaler enabled

In Time Counter mode, when the prescaler is enabled, the output of the prescaler directly clocks Counter (CNR). When LPTMR is enabled, CNR increments every 2^1 to 2^{16} prescaler clock cycles. After LPTMR is enabled, the first increment of CNR takes an additional one or two prescaler clock cycles because of the synchronization logic.

30.3.4.2 Prescaler bypassed

In Time Counter mode, when the prescaler is bypassed, the selected prescaler clock increments Counter (CNR) on every clock cycle. When LPTMR is enabled, the first increment takes an additional one or two prescaler clock cycles because of the synchronization logic.

30.3.4.3 Glitch filter enabled

In Pulse Counter mode, when the glitch filter is enabled, the output of the glitch filter directly clocks Counter (CNR). When LPTMR is first enabled, the output of the glitch filter is asserted, that is, logic 1 for active-high and logic 0 for active-low. The following table shows the change in glitch filter output with the selected input source.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table 169. Glitch filter output with the selected input source

If	Then
The selected input source remains deasserted for at least 2^1 to 2^{15} consecutive prescaler clock rising edges	The glitch filter output also deasserts.
The selected input source remains asserted for at least 2^1 to 2^{15} consecutive prescaler clock rising edges	The glitch filter output also asserts.

NOTE

The input is sampled only on the rising clock edge.

The value of CNR increments each time the glitch filter output asserts. In Pulse Counter mode, the maximum rate at which CNR can increment is once every 2^2 to 2^{16} glitch filter clock edges. When first enabled, the glitch filter waits for an additional one or two glitch filter clock edges because of the synchronization logic.

30.3.4.4 Glitch filter bypassed

In Pulse Counter mode, when the glitch filter is bypassed, the selected input source increments the value of Counter (CNR) every time it asserts. Before LPTMR is first enabled, the selected input source is forced to be asserted. This prevents CNR from incrementing if the selected input source is already asserted when LPTMR is first enabled.

30.3.5 Counter

The value of Counter (CNR) increments by 1 on every:

- Prescaler clock in Time Counter mode, with prescaler bypassed.
- Prescaler output in Time Counter mode, with prescaler enabled.
- Input source assertion in Pulse Counter mode, with glitch filter bypassed.
- Glitch filter output in Pulse Counter mode, with glitch filter enabled.

CNR is reset when LPTMR is disabled or if the counter register overflows. If CSR[TFC] = 0, then CNR is also reset whenever CSR[TCF] = 1.

When the core is halted in Debug mode:

- CNR continues incrementing if configured for Pulse Counter mode.
- CNR stops incrementing if configured for Time Counter mode.

You cannot initialize CNR but can read it at any time. On each read of CNR, you must first write a value to it. This synchronizes and registers the current value of CNR into a temporary register. The contents of the temporary register are returned on each read of CNR.

When reading CNR, the bus clock must be at least two times faster than the rate at which the LPTMR counter is incrementing; otherwise, incorrect data may be returned.

30.3.6 Compare

After the next Counter (CNR) increment (when its value is equal to that of Compare (CMR)), the following events occur:

- CSR[TCF] is read as 1b.
- LPTMR generates an interrupt if CSR[TIE] is 1 as well.
- LPTMR generates a hardware trigger.
- LPTMR writes 0 to CNR if CSR[TFC] is 0.

When LPTMR is enabled, you can modify the value of CMR only when CSR[TCF] is 1. When updating CMR, you must write to it and clear CSR[TCF] before the LPTMR counter increments past the new LPTMR compare value.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

NOTE

When LPTMR is enabled in Time Counter mode, the first increment takes an additional one or two clock cycles because of the synchronization logic. This results in the first compare (and therefore interrupt and hardware trigger) occurring slightly later. A faster prescaler clock or larger prescaler value minimizes this impact.

30.3.7 Interrupt

LPTMR generates an interrupt whenever CSR[TIE] and CSR[TCF] are 1. CSR[TCF] is cleared by disabling LPTMR or writing a logic 1 to it.

You can modify the value of CSR[TIE] and write 1 to CSR[TCF] when LPTMR is enabled.

LPTMR generates an interrupt asynchronously to the system clock. The interrupt can be used to generate a wake-up from any low-power mode, provided LPTMR is enabled as a wake-up source.

30.3.8 Hardware trigger

The LPTMR hardware trigger asserts at the same time CSR[TCF] is set and can be used to trigger hardware events in other peripherals without your intervention. The hardware trigger is always enabled.

Table 170. Hardware trigger

When	Then
CMR[COMPARE] and CSR[TFC] are 0	The LPTMR hardware trigger asserts on the first compare and does not deassert.
CMR[COMPARE] is set to a nonzero value, or if CSR[TFC] = 1	The LPTMR hardware trigger asserts on each compare and deasserts on the following increment of Counter (CNR).

30.4 External signals

Table 171. External signals

Signal	Description		Direction
LPTMR_ALT*n*	Pulse Counter Input LPTMR can select one of the input pins to be used in Pulse Counter mode.		Input
	State meaning	Assertion—If configured for Pulse Counter mode with an active-high input, assertion causes Counter (CNR) to increment. Deassertion—If configured for Pulse Counter mode with an active-low input, deassertion causes CNR to increment.	
	Timing	Assertion or deassertion may occur at any time; input may assert asynchronously to the bus clock.	

30.5 Initialization

Perform the following procedure to initialize LPTMR:

1. Configure Control Status (CSR) for the selected mode and pin configuration, when CSR[TEN] is 0. This resets the counter and clears the flag.
2. Configure Prescaler and Glitch Filter (PSR) with the selected clock source and prescaler or glitch filter configuration.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

3. Configure Compare (CMR) with the selected compare point.
4. Write 1 to CSR[TEN] to enable LPTMR.

30.6 Application information

30.6.1 Application 1: Generate an interrupt every 100 ms using 32.768 kHz clock source

1. Disable LPTMR by writing 0 to CSR[TEN].
2. Select a 32.768 kHz clock source by configuring PSR[PCS].
3. Bypass the prescaler and glitch filter by writing 1 to PSR[PBYP].
4. Assert an interrupt every 3277 cycles by configuring CMR[COMPARE] = 0CCCh.
5. Enable the LPTMR interrupt by writing 1 to CSR[TIE].
6. Enable LPTMR by writing 1 to CSR[TEN].

NOTE

This is just an example. See the chip-specific LPTMR information for the clocks supported on a given chip.

30.6.2 Application 2: Generate an interrupt once a minute using 32.768 kHz clock source

1. Disable LPTMR by writing 0 to CSR[TEN].
2. Select a 32.768 kHz clock source by configuring PSR[PCS].
3. Select the prescaler to divide the prescaler clock by 32768 to increment the counter once a second by configuring PSR[PRESCALE] = 0Eh.
4. Assert an interrupt every 60 seconds by configuring CMR[COMPARE] = 003Bh.
5. Enable the LPTMR interrupt by writing 1 to CSR[TIE].
6. Enable LPTMR by writing 1 to CSR[TEN].

NOTE

This is just an example. See the chip-specific LPTMR information for the clocks supported on a given chip.

30.7 Memory map and register definition

NOTE

The LPTMR registers are reset only on POR only or Cold Reset which would include LVD/HVD sources.. See Reset for more information.

30.7.1 LPTMR register descriptions

30.7.1.1 LPTMR memory map

LPTMR0 base address: 400A_B000h

Offset	Register	Width (In bits)	Access	Reset value
0h	Control Status (CSR)	32	RW	0000_0000h

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Offset	Register	Width (In bits)	Access	Reset value
4h	Prescaler and Glitch Filter (PSR)	32	RW	0000_0000h
8h	Compare (CMR)	32	RW	0000_0000h
Ch	Counter (CNR)	32	RW	0000_0000h

30.7.1.2 Control Status (CSR)

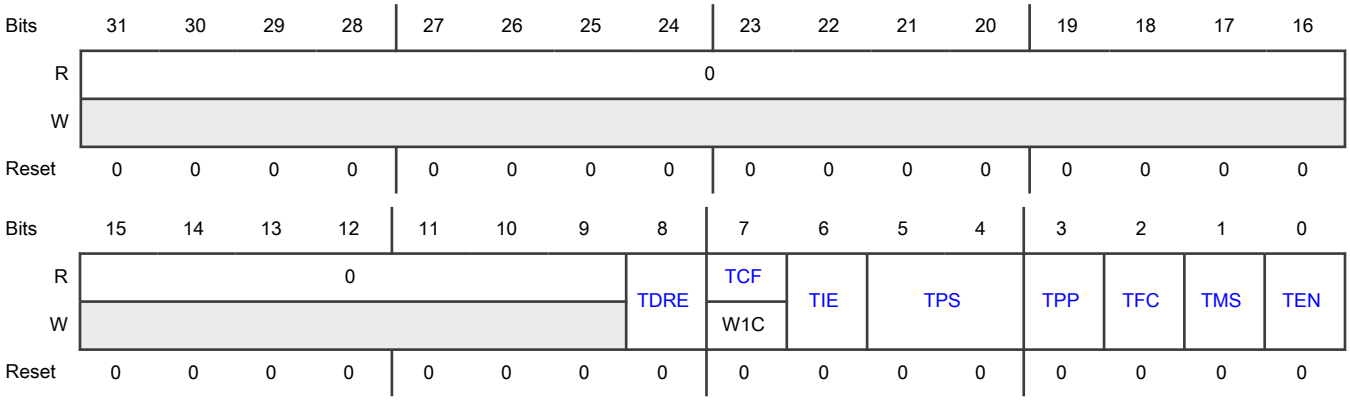
Offset

Register	Offset
CSR	0h

Function

Controls various features of LPTMR.

Diagram



Fields

Field	Function
31-9 —	Reserved
8 TDRE	Timer DMA Request Enable Enables the timer DMA request. When TDRE is 1, the LPTMR DMA request is generated whenever CSR[TCF] is also set. Then, CSR[TCF] is cleared after the DMA controller completes execution. 0b - Disable 1b - Enable

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
7 TCF	Timer Compare Flag Compares the timer. TCF sets on the next Counter (CNR) increment when LPTMR is enabled and Counter (CNR) equals Compare (CMR). TCF is cleared when LPTMR is disabled or a logic 1 is written to it. **NOTE** You must clear this flag before enabling the timer interrupt or DMA request. **NOTE** This field behaves differently for register reads and writes. When reading 0b - CNR ≠ (CMR + 1) 1b - CNR = (CMR + 1) When writing 0b - No effect 1b - Clear the flag
6 TIE	Timer Interrupt Enable Enables the timer interrupt. If TIE is 1, then LPTMR generates an interrupt if CSR[TCF] is 1 as well. 0b - Disable 1b - Enable
5-4 TPS	Timer Pin Select Configures the input source to be used in Pulse Counter mode. The input connections vary by chip. For details, see the chip configuration information about connections to these inputs. You must modify this field only when LPTMR is disabled. 00b - Input 0 01b - Input 1 10b - Input 2 11b - Input 3
3 TPP	Timer Pin Polarity Configures the polarity of the input source in Pulse Counter mode. If TPP is 0, then the pulse counter input source is active-high, and Counter (CNR) increments on the rising edge. If TPP is 1, then the pulse counter input source is active-low, and CNR increments on the falling edge. You must modify this field only when LPTMR is disabled. 0b - Active-high 1b - Active-low

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
2 TFC	Timer Free-Running Counter Specifies when the counter resets. If TFC is 0, Counter (CNR) resets on the count cycle following Counter (CNR) becoming equal to Compare (CMR). If TFC is 1, CNR resets on overflow. In both cases, CSR[TCF] sets to 1 on the cycle after CNR and CMR match. You must modify this field only when LPTMR is disabled. 0b - Reset when TCF asserts 1b - Reset on overflow
1 TMS	Timer Mode Select Configures the mode of LPTMR. You must modify this field only when LPTMR is disabled. 0b - Time Counter 1b - Pulse Counter
0 TEN	Timer Enable Enables the LPTMR timer. If TEN is 0, it resets the LPTMR internal logic, including CNR[COUNTER] and CSR[TCF]. If TEN is 1, LPTMR is enabled. Do not alter CSR[5:1] when writing 1 to this field. 0b - Disable 1b - Enable

30.7.1.3 Prescaler and Glitch Filter (PSR)

Offset

Register	Offset
PSR	4h

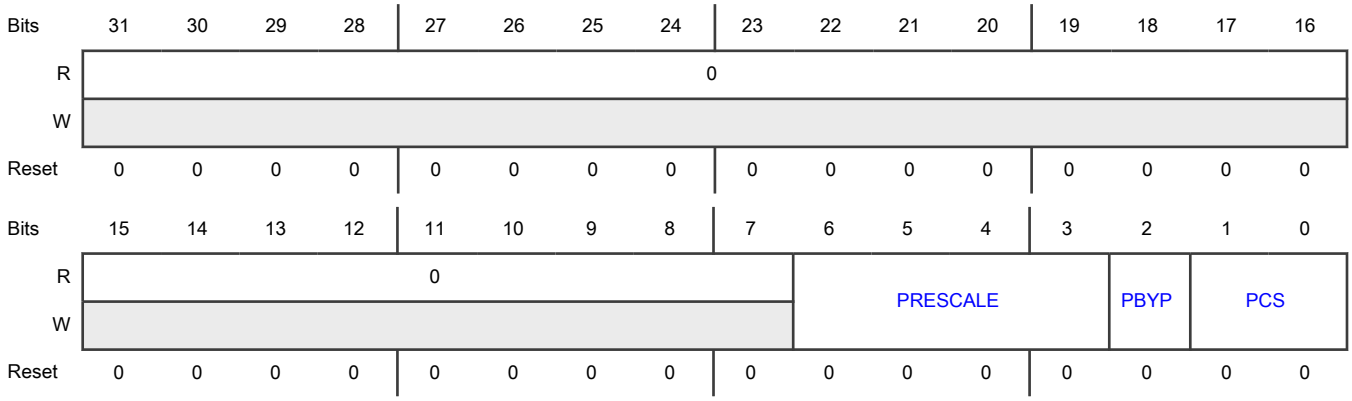
Function

Configures features related to the prescaler and glitch filter.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram



Fields

Field	Function
31-7 —	Reserved
6-3 PRESCALE	Prescaler and Glitch Filter Value Configures the size of the prescaler in Time Counter mode and the width of the glitch filter in Pulse Counter mode. The width of the glitch filter can vary by one cycle because of the pulse counter input synchronization. You must modify this field only when LPTMR is disabled. 0000b - Prescaler divides the prescaler clock by 2; glitch filter does not support this configuration 0001b - Prescaler divides the prescaler clock by 4; glitch filter recognizes change on input pin after two rising clock edges 0010b - Prescaler divides the prescaler clock by 8; glitch filter recognizes change on input pin after four rising clock edges 0011b - Prescaler divides the prescaler clock by 16; glitch filter recognizes change on input pin after eight rising clock edges 0100b - Prescaler divides the prescaler clock by 32; glitch filter recognizes change on input pin after 16 rising clock edges 0101b - Prescaler divides the prescaler clock by 64; glitch filter recognizes change on input pin after 32 rising clock edges 0110b - Prescaler divides the prescaler clock by 128; glitch filter recognizes change on input pin after 64 rising clock edges 0111b - Prescaler divides the prescaler clock by 256; glitch filter recognizes change on input pin after 128 rising clock edges 1000b - Prescaler divides the prescaler clock by 512; glitch filter recognizes change on input pin after 256 rising clock edges 1001b - Prescaler divides the prescaler clock by 1024; glitch filter recognizes change on input pin after 512 rising clock edges

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	1010b - Prescaler divides the prescaler clock by 2048; glitch filter recognizes change on input pin after 1024 rising clock edges 1011b - Prescaler divides the prescaler clock by 4096; glitch filter recognizes change on input pin after 2048 rising clock edges 1100b - Prescaler divides the prescaler clock by 8192; glitch filter recognizes change on input pin after 4096 rising clock edges 1101b - Prescaler divides the prescaler clock by 16,384; glitch filter recognizes change on input pin after 8192 rising clock edges 1110b - Prescaler divides the prescaler clock by 32,768; glitch filter recognizes change on input pin after 16,384 rising clock edges 1111b - Prescaler divides the prescaler clock by 65,536; glitch filter recognizes change on input pin after 32,768 rising clock edges
2 PBYP	Prescaler and Glitch Filter Bypass Controls the clocking of Counter (CNR). If PBYP is 0, the output of the prescaler or glitch filter clocks CNR. If PBYP is 1, the selected prescaler clock in Time Counter mode, or else the selected input source in Pulse Counter mode, directly clocks CNR. You must modify this field only when LPTMR is disabled. 0b - Prescaler and glitch filter enable 1b - Prescaler and glitch filter bypass
1-0 PCS	Prescaler and Glitch Filter Clock Select Selects the clock to be used by the LPTMR prescaler and glitch filter. In Time Counter mode, PCS selects the input clock to the prescaler. In Pulse Counter mode, PCS selects the input clock to the glitch filter. See the chip configuration details for information on connections to these inputs. You must modify this field only when LPTMR is disabled. 00b - Clock 0 01b - Clock 1 10b - Clock 2 11b - Clock 3

30.7.1.4 Compare (CMR)

Offset

Register	Offset
CMR	8h

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Function

Configures the compare values to Counter (CNR) (see Compare for more information).

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	COMPARE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	COMPARE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-0 COMPARE	Compare Value Configures the compare values to Counter (CNR). On the next CNR increment, if LPTMR is enabled and Counter (CNR) equals Compare (CMR), then: 1. LPTMR writes 1 to CSR[TCF]. 2. The hardware trigger asserts until the next time CNR increments. If CMR = 0, the hardware trigger remains asserted until LPTMR is disabled. If LPTMR is enabled, you must modify the value of CMR only if CSR[TCF] is 1.

30.7.1.5 Counter (CNR)

Offset

Register	Offset
CNR	Ch

Function

Specifies counter values (see Counter for more information).

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	COUNTER															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	COUNTER															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-0 COUNTER	Counter Value Contains the current value of the LPTMR counter at the time you last wrote to this register.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Chapter 31
Low Power Inter-Integrated Circuit (LPI2C)

31.1 Chip-specific LPI2C information

Table 172. Reference links to related information

Topic	Related module	Reference
Full description	LPI2C	LPI2C
System memory map		Memory map
Clocking		Clock distribution
Power Management		Power Management
Signal multiplexing	Port control	Signal multiplexing

31.1.1 Module instances

This chip contains one instance of LPI2C: LPI2C0.

31.2 Overview

LPI2C supports an efficient interface to an I2C bus as a controller and target:

- Implements logic support for Standard, Fast, Fast+, HS-mode (target only) and Ultra-Fast modes of operation
- Uses little CPU overhead, with DMA offloading of FIFO register accesses
- Continues operating in Deep Sleep modes if an appropriate clock is available

LPI2C also complies with the System Management Bus (SMBus) Specification, version 3. The SMBus is a single-ended simple two-wire bus, which is typically used for low-bandwidth communications.

The Inter-Integrated Circuit (I^2C) serial bus is multi-controller, multi-target, packet-switched, and single-ended, and is often used to attach microcontroller ICs to lower-speed peripheral ICs.

NOTE

Terminology in this chapter has been updated to align with I^2C-bus specification, Rev. 7.0, as shown in Table 173.

Table 173. Updated terms

Updated term	Deprecated term
Controller	Master
Target	Slave

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

31.2.1 Block diagram

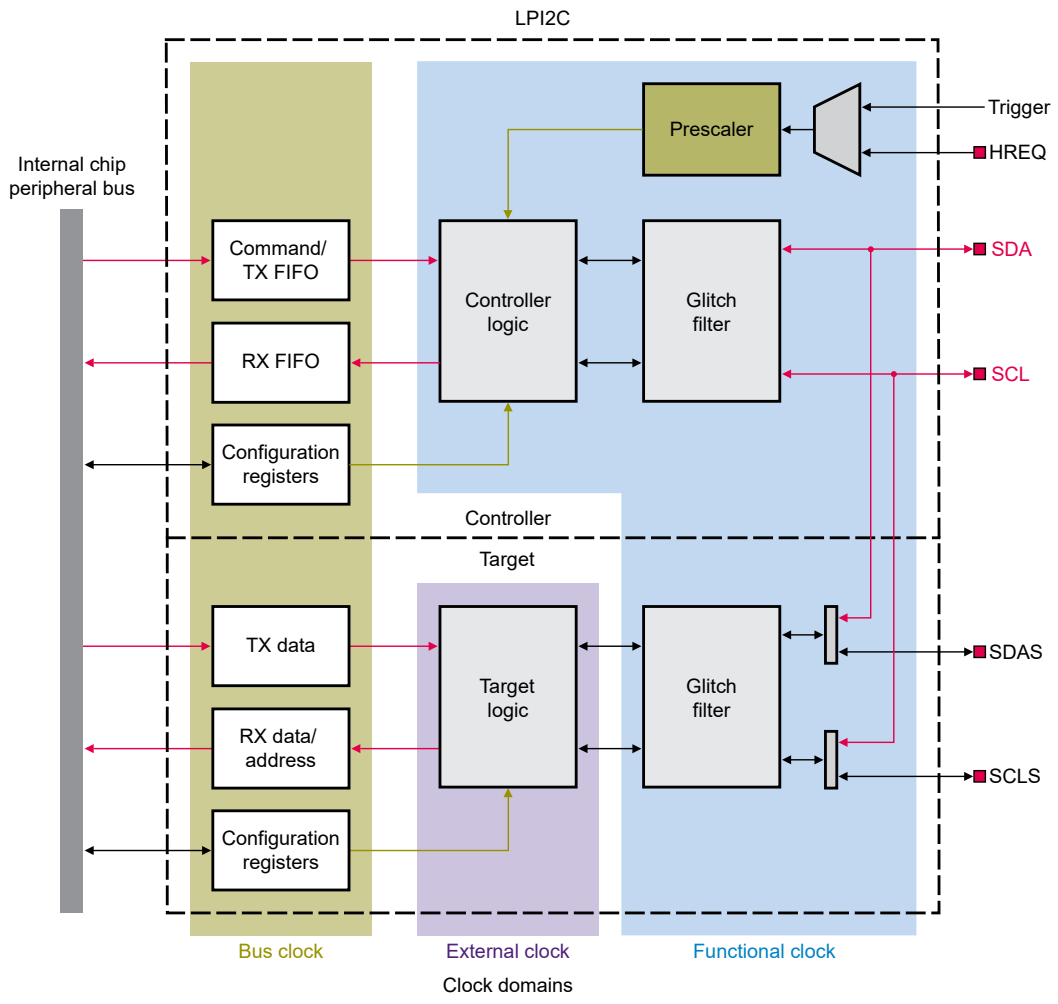


Figure 131. Block diagram

31.2.2 Features

LPI2C supports:

- Standard, Fast, Fast+, HS (target only), and Ultra Fast modes
- Multicontroller, including synchronization and arbitration, means that any number of controller nodes can be present. Also, controller and target roles can be changed between messages (after a Stop signal is sent).
- Clock stretching. Used on the SCL line, as an I2C flow control mechanism.
- Arbitration, used when the system has more than one controller, ensures that only one I2C transmitter is active at a time.
- General call, seven-bit addressing, and ten-bit addressing
- Software reset, Start byte, and device ID (also require software support)

The LPI2C controller supports:

- Command and transmit FIFO of 4 words (8-bit transmit data + 3-bit command)
- Receive FIFO of 4 words (8-bit receive data).
- Command FIFO waiting for an I2C idle bus before initiating a transfer

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

- Initiation of repeated Start and Stop conditions and one or more controller-receiver transfers by command FIFO
- Stop condition generation from command FIFO, or automatic generation of Stop condition when the transmit FIFO is empty
- Host request input to control the start time of an I2C bus transfer
- Interrupt generation on data match and unwanted data rejection, via flexible receive data match
- Flags and optional interrupt signals at repeated Start condition, Stop condition, loss of arbitration, unexpected NACK, and command word errors
- Configurable bus idle timeout and pin-stuck-low timeout

The LPI2C target supports:

- Separate I2C target registers to minimize software overhead because of controller or target switching
- 7-bit or 10-bit addressing, address range, SMBus alert, and general call address.
- Transmit data register that supports interrupt or DMA requests
- Receive data register that supports interrupt or DMA requests
- Software-controllable ACK or NACK, with optional clock stretching on ACK or NACK field
- Configurable clock stretching, to avoid transmit-FIFO-underrun and receive-FIFO-overrun errors
- Flags and optional interrupt at end of packet, Stop condition, or bit error detection

31.3 Functional description

31.3.1 Controller mode

The LPI2C controller logic operates independently from the target logic to perform all controller-mode transfers on the I2C bus.

31.3.1.1 Transmit and Command FIFO commands

The transmit FIFO stores command data to initiate various I2C operations. The following operations can be initiated through commands in the transmit FIFO:

- Start or repeated Start condition with address byte, expecting ACK or NACK.
- Transmit data. This operation is the default for zero-extended-byte writes to the transmit FIFO.
- Receive 1-256 bytes of data. You can configure this operation to discard received data and not to store it in the receive FIFO.
- Stop condition. You can configure this operation to send a Stop condition when the transmit FIFO is empty.

Multiple transmit and receive commands can be inserted between the Start and Stop conditions. To comply with the I2C specification, transmit and receive commands must not be interleaved. The receive data command and the receive data and discard commands can be interleaved. This interleaving ensures that only the desired received data is stored in the receive FIFO (or compared with the data match logic).

The LPI2C controller automatically transmits a NACK on the last byte of a receive data command. It transmits the NACK unless the next command in the FIFO is also a receive data command. If the transmit FIFO is empty when a receive data command completes, a NACK is also automatically transmitted.

The LPI2C controller supports 10-bit addressing via a (repeated) Start condition, followed by a transmit data byte containing the second address byte, followed by any number of data bytes with the controller transmit data.

A Start or repeated Start condition expecting a NACK (for example, HS mode controller code) must be followed by a Stop or (repeated) Start condition.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

31.3.1.2 Controller operations

When LPI2C is enabled, it monitors the I2C bus to detect when the I2C is idle (MSR[BBF]). If either SCL or SDA are low, the I2C bus is no longer considered idle. The bus becomes idle if a Stop condition is detected or if a bus idle timeout is detected (as configured by MCFGR2[BUSIDLE]). After the bus is idle, if the transmit FIFO is not empty and the host request is asserted or disabled, the LPI2C controller initiates a transfer on the bus. This transfer involves the following steps:

1. Wait the bus idle time equal to (MCCR0[CLKLO] + 1) multiplied by the prescaler (MCFGR1[PRESCALE]).
2. Transmit a Start condition and address byte using the timing configuration in Controller Clock Configuration 0 (MCCR0). If an HS mode transfer is configured, the timing configuration from Controller Clock Configuration 1 (MCCR1) is used instead.
3. Perform controller transmit or controller receive transfers, as configured by the transmit FIFO.
4. Transmit NACK on the last byte of a controller receive transfer. This action is performed unless the next command in the transmit FIFO is also a receive data command and the transmit FIFO is not empty.
5. Transmit a repeated Start or Stop condition as configured by the transmit FIFO or MCFGR1[AUTOSTOP]. A repeated Start can change which timing configuration register is used.

When MCFGR0[RELAX] is 1, the LPI2C controller does not wait for the I2C bus to be idle before starting a transfer. It also relaxes some restrictions on the order of FIFO commands. For example, it allows a Stop command when it is idle to generate a Start condition, followed by one SCL clock pulse, and then a Stop condition.

When the LPI2C controller is disabled, LPI2C continues emptying the transmit FIFO until a Stop condition is transmitted. (The controller could be disabled due to MCR[MEN] being 0, or automatically due to mode entry.) However, LPI2C no longer stalls the I2C bus by waiting for the transmit or receive FIFO. After the transmit FIFO is empty, LPI2C generates a Stop condition automatically.

The LPI2C controller can stall the I2C bus under certain conditions. This stalling results in SCL pulled low continuously on the first bit of a byte, until these conditions change:

- The LPI2C controller is enabled and busy, the transmit FIFO is empty, and MCFGR1[AUTOSTOP] is 0. The LPI2C controller continues to stall the bus until the transmit FIFO is loaded with more data.
- The LPI2C controller is enabled and receiving data, receive data is not being discarded (due to command or receive data match), and the receive FIFO is full. The LPI2C controller continues to stall the I2C bus until the receive FIFO is emptied.

The LPI2C controller aborts the existing transfer when MCFGR0[ABORT] becomes 1. This action causes the LPI2C controller to complete the existing word and then transmit a Stop condition. If the receive FIFO is full when the receive operation is aborted, the last received data is discarded. A new transfer does not start while MCFGR0[ABORT] is 1.

31.3.1.3 Receive FIFO and data matching

The receive FIFO stores receive data during controller-receiver transfers. You can configure the LPI2C controller to discard received data instead of storing it in the receive FIFO. This option is configured via the command word in the transmit FIFO.

Received data supports a receive data match function that can match received data against one of two bytes, or against a masked data byte. You can configure the data match function to compare only the first one or two data words received since the last (repeated) Start condition. Received data that is already discarded due to the command word cannot cause a data match. It delays the match on the first data word received until after the discarded data is received.

You can configure the receiver match function to discard all received data until a data match is detected, using MCFGR0[RDMO]. Following a data match, write 0 to MCFGR0[RDMO] before writing 0 to MSR[DMF] to allow all subsequent data to be received.

31.3.1.4 Timing parameters

The LPI2C controller can configure the following timing parameters. Parameters are configured separately for HS mode (Controller Clock Configuration 1 (MCCR1)) and other modes (Controller Clock Configuration 0 (MCCR0)). This separation allows the HS mode controller code to be sent using regular timing parameters. Then it allows a switch to HS mode timing (following a repeated Start) until the next STOP condition.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Configure the LPI2C controller timing parameters, measured in LPI2C functional clock cycles, as shown in Table 174. You must configure these parameters to meet the I2C timing specification for the required mode.

Table 174. Timing parameters

I2C specification timing parameter	I2C specification timing symbol	LPI2C timing parameter (in LPI2C functional clock cycles)
SCL clock period	tSCL	(CLKHI + CLKLO + 2 + SCL_LATENCY) × (2 ^ PRESCALE)
Hold time (repeated) Start condition	tHD:STA	(SETHOLD +1) × (2 ^ PRESCALE)
Low period of the SCL clock	tLOW	(CLKLO + 1) × (2 ^ PRESCALE)
High period of the SCL clock	tHIGH	(CLKHI + 1 + SCL_LATENCY) × (2 ^ PRESCALE)
Setup time for a repeated Start condition or Stop condition	tSU:STA, tSU:STO	(SETHOLD + 1 + SCL_LATENCY) × (2 ^ PRESCALE)
Data hold time	tHD:DAT	(DATAVD + 1) × (2 ^ PRESCALE)
Data setup time	tSU:DAT	(SDA_LATENCY + 1) × (2 ^ PRESCALE)
Bus free time between a Stop and Start condition	tBUF	(CLKLO + 1 + SDA_LATENCY) × (2 ^ PRESCALE)
Data valid time, data valid acknowledge time	tVD:DAT, tVD:ACK	(DATAVD + 1) × (2 ^ PRESCALE)

Table 175 defines the latency parameters. These parameters assume that the rise time is less than one LPI2C functional clock cycle. The rise time depends on a number of factors, including the I/O propagation delay, the I2C bus loading, and the external pullup resistor sizing. A larger rise time increases the number of cycles that the signal takes to propagate through the synchronizer (and glitch filter), which increases the latency.

Table 175. Synchronization latency

Timing parameter	Timing definition
SCL_LATENCY	ROUNDDOWN ((2 + FILTSCL + SCL_RISETIME) ÷ (2 ^ PRESCALE))
SDA_LATENCY	ROUNDDOWN ((2 + FILTSDA + SDA_RISETIME) ÷ (2 ^ PRESCALE))

The following timing restrictions must be enforced to avoid unexpected Start or Stop conditions on the I2C bus. These restrictions also avoid unexpected Start or Stop conditions detected by the LPI2C controller. The timing restrictions can be summarized as "SDA cannot change when SCL is high outside a transmitted (repeated) Start or Stop condition."

Table 176. LPI2C timing parameter restrictions

Timing parameter	Minimum	Maximum	Comment
CLKLO	03h	—	CLKLO x (2 ^ PRESCALE) > SCL_LATENCY
CLKHI	01h	—	Configure CLKHI to meet the duty cycle requirements in the I2C specification
SETHOLD	02h	—	SETHOLD × (2 ^ PRESCALE) > SDA_LATENCY

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table 176. LPI2C timing parameter restrictions (continued)

Timing parameter	Minimum	Maximum	Comment
DATAVD	01h	CLKLO – SDA_LATENCY – 1	Configure DATAVD to meet the data hold requirement in the I2C specification
FILTSCL	00h	[CLKLO × (2 ^ PRESCALE)] – 3	FILTSCL and FILTSDA are the only parameters not multiplied by (2 ^ PRESCALE)
FILTSDA	FILTSCL	[CLKLO × (2 ^ PRESCALE)] – 3	Configuring FILTSDA greater than FILTSCL can delay the SDA input to compensate for board level skew
BUSIDLE	(CLKLO + SETHOLD + 2) × 2	—	Must also be greater than (CLKHI + 1)

See the UM10204, *I2C-bus specification and user manual*.

See Application information for example LPI2C timing configurations.

31.3.1.5 Error conditions

The LPI2C controller monitors errors while it is active. The following conditions generate an error flag and block a new Start condition from being sent, until the flag is cleared by software:

- A Start or Stop condition is detected and is not generated by the LPI2C controller (MSR[ALF] becomes 1).
- Transmitting data on SDA and different values are received (MSR[ALF] becomes 1). LPI2C controller stops driving SDA immediately, but continues to drive SCL for the remainder of the byte.
- NACK is detected when transmitting data, and MCFGR1[IGNACK] is 0 (MSR[NDF] becomes 1).
- NACK is detected and is expecting ACK for the address byte, and MCFGR1[IGNACK] is 0 (MSR[NDF] becomes 1).
- ACK is detected and is expecting NACK for the address byte, and MCFGR1[IGNACK] is 0 (MSR[NDF] becomes 1).
- Transmit FIFO is requesting to transmit or receive data without a Start condition (MSR[FEF] becomes 1). This restriction is ignored when MCFGR0[RELAX] is 1.
- SCL (or SDA if MCFGR1[TIMECFG] is 1) is low for (MCFGR2[TIMELOW] × 256) prescaler cycles without a pin transition (MSR[PLTF] becomes 1).

You must respond to MSR[PLTF] to terminate the existing command. You can terminate the command cleanly by writing 0 to MCR[MEN], or you can terminate it abruptly by writing 1 to MCR[RST].

You can use MCFGR0[RELAX] to attempt to recover a target with SDA stuck low. If MCFGR0[RELAX] is 1, the LPI2C controller does not wait for the I2C bus to be idle before starting a transfer. Initiating a Start command with address FFh, then the Stop condition, should generate sufficient SCL clock edges to cause the target to release SDA.

You can use MCFGR2[BUSIDLE] to force the I2C bus to be considered idle when SCL and SDA remain high for (BUSIDLE + 1) prescaler cycles. The bus is considered idle when the LPI2C controller is first enabled. When BUSIDLE is configured greater than zero, then SCL or SDA must be high for (BUSIDLE + 1) prescaler cycles before the I2C bus is considered idle.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

31.3.1.6 Pin configuration

Configuration	Description
Open-drain support	The LPI2C controller defaults to open-drain configuration of the SDA and SCL pins. Support for true open drain depends on the specific device, and requires the pins where LPI2C pins are muxed to support true open drain.
HS mode support	Support for HS mode depends on the specific device. This mode requires the SCL pin to support the current source pullup required in the I2C specification.
Ultra-Fast mode support	The LPI2C controller supports the output-only push-pull function required for I2C Ultra-Fast mode using the SDA and SCL pins. Support for Ultra-Fast mode also requires MCFGR1[IGNACK] to be 1.
Push-pull two-wire support	A push-pull two-wire configuration is available to the LPI2C controller. If LPI2C is the only controller and all I2C pins on the bus are at the same voltage, this configuration may support a partial HS mode. A partial HS mode, not a full HS mode, because this configuration actively drives high rather than enabling a current service pull-up. This configuration sets the SCL pin as push-pull for every clock except the first clock pulse, to allow HS-mode-compatible targets to perform clock stretching. In this mode, the SDA pin is tristated for controller-receive data bits and controller-transmit ACK/NACK bits, and is configured as push-pull at other times. To avoid the risk of contention when SDA is push-pull, configure the pin for open-drain operation, as part of the device-specific configuration.
Push-pull four-wire support	The push-pull four-wire configuration separates the SCL input data and output data into separate pins. It also separates the SDA input data and output data into separate pins. The SCL/SDA pins are used for input data; the SCLS/SDAS pins are used for output data, with configurable polarity. This configuration simplifies external connections when connecting the LPI2C to the I2C bus through external level shifters or discrete components. When using this four-wire configuration, the LPI2C controller logic and LPI2C target logic cannot connect to separate I2C buses.

31.3.2 Target mode

To perform all target mode transfers on the I2C bus, the LPI2C target logic operates independently from the LPI2C controller logic.

31.3.2.1 Address matching

You can configure the LPI2C target:

- To match one of two addresses, using either 7-bit or 10-bit addressing modes for each address.
- To match a range of addresses in either 7-bit or 10-bit addressing modes.
- To match the general call address and generate appropriate flags.
- To match the SMBus alert address and generate appropriate flags.
- To detect the HS mode controller code address, and to disable the digital filters and output valid delay time until the next Stop condition is detected.

After a valid address is matched, the LPI2C target automatically performs target-transmit or target-receive transfers until:

- A NACK is detected (unless SCFGR1[IGNACK] becomes 1).
- A bit error is detected (the LPI2C target is driving SDA, but a different value is sampled).
- A (repeated) Start or Stop condition is detected.

When SCFGR1[RXALL] is 1, the LPI2C target receives all addresses and data on the I2C bus. Unless the address is matched, it never drives the SDA pin. Configure the LPI2C target to match only on 7-bit address mode when this feature is enabled. Software can support 10-bit address mode. This option is intended for debugging the contents of an I2C transfer between another controller

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

and target. Likewise, SCFGR1[SDCFG] and SCFGR1[RSCFG] configure the Stop or repeated Start flags to assert on all Stop or repeated Start conditions, even when there is no address match.

31.3.2.2 Transmit and receive data

Target Transmit Data (STDR) and Target Receive Data (SRDR) are double-buffered and only update during a target-transmit and target-receive transfer, respectively.

You can configure the target address that was received to be read from SRDR (for example, when using DMA to transfer data) or from Target Address Status (SASR).

You can configure STDR to request data only after a target-transmit transfer is detected. You can also configure it to request new data whenever STDR is empty.

Write to STDR only when SSR[TDF] is set.

Read SRDR only when SSR[RDF] is set, or when SSR[AVF] is set and SCFGR1[RXCFG] = 1.

Read SASR only when SSR[AVF] is set.

31.3.2.3 Read request

The LPI2C target generates a read request when the I2C bus is idle and you write 1 to SCFGR0[RDREQ]. This occurrence causes the LPI2C target to pull the SDA pin low until either the first SCL falling edge or SCFGR0[RDREQ] becomes 0. Following the next Stop condition or a software timeout, you can proceed as follows:

- If the LPI2C target is accessed at the correct address, the read request is acknowledged, and you must write 0 to SCFGR0[RDREQ].
- If SCFGR0[RDACK] is 1, the read request is acknowledged by a Start followed by one SCL pulse followed by a STOP condition. You must write 0 to SCFGR0[RDREQ].
- If neither of the first two cases occurred, the read request is not acknowledged. You must write 0 and then 1 to SCFGR0[RDREQ] after an appropriate delay, provided the I2C bus is idle.
- If a software timeout occurs, the read request is not acknowledged, and you should write 0 to SCFGR0[RDREQ]. Another request can be generated after an appropriate delay.

Writing to SCFGR0[RDREQ] when the I2C bus is busy results in a logic 0 being written. Write 1 to SCFGR0[RDREQ], then confirm that the register is written to correctly. If the register does not update correctly, the I2C bus is no longer idle and the read request must be attempted later.

31.3.2.4 Clock stretching

The LPI2C target supports many configurable options for clock stretching. You can configure these conditions to perform clock stretching:

- SSR[AVF] is set during the ninth clock pulse of the address byte.
- SSR[TDF] is set during the ninth clock pulse of a target-transmit transfer.
- SSR[RDF] is set during the ninth clock pulse of a target-receive transfer.
- SSR[TAF] is set during the eighth clock pulse of an address byte or a target-receive transfer. In HS mode, this option is disabled.
- Clock stretching can be extended for a number of cycles equal to the value of SCFGR2[CLKHOLD] cycles. This stretching allows additional setup time to sample the SDA pin externally. In HS mode, this option is disabled.

When clock stretching is enabled, clock stretching extends for one peripheral bus clock cycle after SDA updates, unless extended by the SCFGR2[CLKHOLD] configuration.

Unless configured to 0, clock stretching can be extended for a number of cycles equal to the value of SCFGR2[CLKHOLD] +3 cycles.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

31.3.2.5 Timing parameters

The LPI2C target can configure the following timing parameters:

- SDA data valid time from SCL negation to SDA update
- SCL hold time when clock stretching is enabled to increase setup time when sampling SDA externally
- SCL glitch filter time
- SDA glitch filter time

These parameters are disabled when SCR[FILTEN] is 0, when SCR[FILTDZ] is 1 in Doze mode, and when LPI2C target detects HS mode. When disabled, the LPI2C target is clocked directly from the I2C bus. In this case, the target may not satisfy all timing requirements of the I2C specification (such as SDA minimum hold time in Standard/Fast mode).

The LPI2C target places the following restrictions on the timing parameters:

- You must configure SCFGR2[FILTSDA] to be greater than or equal to SCFGR2[FILTSCL] (unless compensating for board level skew between SDA and SCL).
- You must configure SCFGR2[DATAVD] to be less than the minimum SCL low period.

31.3.2.6 Error conditions

The LPI2C target can flag the following error conditions:

- SSR[BEF] is set when the LPI2C target is driving SDA but it samples a different value than what is expected.
- SSR[FEF] is set due to a transmit data underrun or a receive data overrun. To eliminate the possibility of underrun and overrun, enable clock stretching. When SCFGR1[RXNACK] is 1, the LPI2C target generates a NACK on a receive data overrun.
- SSR[FEF] is also set due to an address overrun, but only when SCFGR1[RXCFG] is 1. To eliminate the possibility of overrun, enable clock stretching. When SCFGR1[RXNACK] is 1, the LPI2C target generates a NACK on an address overrun regardless of the value of SCFGR1[RXCFG].

The LPI2C target does not implement a timeout due to SCL or SDA being stuck low. If this detection is required, use the LPI2C controller logic so you can reset the LPI2C target when this condition is detected.

31.3.3 Low-power modes

LPI2C remains functional during low-power modes, if MCR[DOZEN] = 0 and LPI2C uses an external or internal clock source that remains enabled. LPI2C can generate an interrupt or DMA request to cause a wake-up from low-power modes.

You can configure LPI2C to be disabled in low-power modes when MCR[DOZEN] = 1. In this case, LPI2C waits for the current transfer to complete any pending operation.

NOTE

See the chip-specific information for low-power modes available on your chip.

31.3.4 Debug mode

Table 177. Debug mode

Mode	LPI2C operation
Debug	If MCR[DBGEN] = 1, can continue operating in Debug mode.

31.3.5 Peripheral triggers

The connection of the LPI2C peripheral triggers to other peripherals depends upon the specific device being used.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table 178. LPI2C triggers

Trigger	Description
Controller output trigger	Generates an output trigger that can be connected to other peripherals on the device. The controller output trigger asserts on either a repeated Start or a Stop condition. The trigger remains asserted for one cycle of the LPI2C functional clock divided by MCFGR1[PRESCALE].
Target output trigger	Generates an output trigger that can be connected to other peripherals on the device. The target output trigger asserts on either a repeated Start or a Stop condition that occurs after a target address match. The target output trigger remains asserted until the next target SCL pin negation.
Input trigger	To control the start of a LPI2C bus transfer, the LPI2C input trigger can be selected instead of the HREQ input. The input trigger is synchronized. To be detected, the input trigger must assert for at least two cycles of the LPI2C functional clock divided by the value of MCFGR1[PRESCALE]. When LPI2C is busy, the HREQ input (and therefore the input trigger) is ignored.

31.3.6 Clocking

Table 179. LPI2C clocks

Clock	Description
LPI2C functional clock	The LPI2C functional clock is asynchronous to the bus clock. It can remain enabled in low-power modes to support I2C bus transfers by the LPI2C controller. The functional clock is also used by the LPI2C target to support digital filter and data hold time configurations. The LPI2C controller divides the functional clock by a prescaler (MCFGR1[PRESCALE]) and the resulting frequency must be at least eight times faster than the I2C bus bandwidth.
External clock	The LPI2C target logic is clocked directly from the external pins. These pins are SCL and SDA, or SCLS and SDAS if the controller and target are implemented on separate pins). This clocking allows the LPI2C target to remain operational, even when the LPI2C functional clock is disabled. **NOTE** If the LPI2C functional clock is disabled, the LPI2C target digital filter must be disabled. This condition can affect compliance with some timing parameters of the I2C specification, such as data hold time.
Bus clock	The bus clock is only used for bus accesses to the control and configuration registers. The bus clock frequency must be sufficient to support the data bandwidth requirements of the LPI2C controller and target registers.

For chip-specific clocking information, see the Clocking chapter.

31.3.7 Reset

Table 180. LPI2C resets

Reset	Description
Chip reset	The logic and registers for the LPI2C controller and target are reset to their default states after a chip reset.
Software reset	The LPI2C controller implements a software reset field in its control register. MCR[RST] resets all controller logic and registers to their default states, except for Controller Control (MCR) itself.

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table 180. LPI2C resets (continued)

Reset	Description
	The LPI2C target implements a software reset field in its control register. SCR[RST] resets all target logic and registers to their default states, except for Target Control (SCR) itself.
FIFO reset	The LPI2C controller implements write-only control fields that reset the transmit FIFO (MCR[RTF]) and receive FIFO (MCR[RRF]). After a FIFO is reset, that FIFO is empty. The LPI2C target implements write-only control fields that reset the transmit data register (SCR[RTF]) and receive data register (SCR[RRF]). After a data register is reset, that data register is empty.

31.3.8 Interrupts and DMA requests

Depending on the configuration, interrupts and DMA requests can be combined:

- LPI2C controller and target interrupts
- LPI2C controller and target transmit DMA requests
- LPI2C controller and target receive DMA requests

31.3.8.1 Controller mode

Table 181 lists the Controller mode sources that can generate LPI2C controller interrupts and LPI2C controller transmit and receive DMA requests.

Table 181. Controller interrupts and DMA requests

Status flag	Description	Can generate		
		Interrupt?	DMA request?	Low-power wake-up?
Transmit Data Flag (MSR[TDF])	Data can be written to transmit FIFO, as configured by MFCR[TXWATER].	Y	TX	Y
Receive Data Flag (MSR[RDF])	Data can be read from the receive FIFO, as configured by MFCR[RXWATER].	Y	RX	Y
End Packet Flag (MSR[EPF])	Controller has transmitted a repeated Start or Stop condition.	Y	N	Y
Stop Detect Flag (MSR[SDF])	Controller has transmitted a Stop condition (and the LPI2C controller is idle, if MCFGR1[STOPCFG] = 1) .	Y	N	Y
NACK Detect Flag (MSR[NDF])	During an address byte, the controller expects an ACK but detects a NACK. During an address byte, the controller expects a NACK but detects an ACK. During a controller-transmitter data byte, the controller detects a NACK.	Y	N	Y
Arbitration Lost Flag (MSR[ALF])	The controller lost arbitration due to a Start or Stop condition detected at the wrong time, or the controller was transmitting data	Y	N	Y

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table 181. Controller interrupts and DMA requests (continued)

Status flag	Description	Can generate		
		Interrupt?	DMA request?	Low-power wake-up?
	but received data different from the data that was transmitted.			
FIFO Error Flag (MSR[FEF])	The controller expects a Start condition in the command FIFO, but the next entry in the command FIFO is not a Start condition.	Y	N	Y
Pin Low Timeout Flag (MSR[PLTF])	Pin low timeout is enabled and SCL (or SDA, if configured) is low for longer than the configured timeout.	Y	N	Y
Data Match Flag (MSR[DMF])	The received data matches the configured data match, but the received data is not discarded due to a command FIFO entry.	Y	N	Y
Start Detect Flag (MSR[STF])	A Start condition is detected on the I2C bus (and the LPI2C controller is idle, if MCFGR1[STARTCFG] = 1).	Y	N	Y
Controller Busy Flag (MSR[MBF])	LPI2C controller is busy transmitting or receiving data.	N	N	N
Bus Busy Flag (MSR[BBF])	LPI2C controller is enabled and activity is detected on the I2C bus, but no Stop condition is detected and no bus idle timeout (if enabled) occurred.	N	N	N

31.3.8.2 Target mode

Table 182 lists the target mode sources that can generate LPI2C target interrupts and LPI2C target transmit and receive DMA requests.

Table 182. Target interrupts and DMA requests

Status flag	Description	Can generate		
		Interrupt?	DMA request?	Low-power wake-up?
Transmit Data Flag (SSR[TDF])	Data can be written to Target Transmit Data (STDR).	Y	TX	Y
Receive Data Flag (SSR[RDF])	Data can be read from Target Receive Data (SRDR).	Y	RX	Y
Address Valid Flag (SSR[AVF])	Address can be read from Target Address Status (SASR).	Y	RX	Y
Transmit ACK Flag (SSR[TAF])	ACK or NACK can be written to Target Transmit ACK (STAR).	Y	N	Y
Repeated Start Flag (SSR[RSF])	Target has detected an address match followed by a repeated Start condition.	Y	RX	Y

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table 182. Target interrupts and DMA requests (continued)

Status flag	Description	Can generate		
		Interrupt?	DMA request?	Low-power wake-up?
Stop Detect Flag (SSR[SDF])	Target has detected an address match followed by a Stop condition.	Y	RX	Y
Bit Error Flag (SSR[BEF])	Target was transmitting data, but received data is different from what was transmitted.	Y	N	Y
FIFO Error Flag (SSR[FEF])	This flag is set by: • Transmit data underrun • Receive data overrun • Address status overrun when SCFGR1[RXCFG] = 1 This flag can only be set when clock stretching is disabled.	Y	N	Y
Address Match 0 Flag (SSR[AM0F])	Target detected an address match SAMR[ADDR0].	Y	N	N
Address Match 1 Flag (SSR[AM1F])	Target detected an address match with SAMR[ADDR1] or using an address range.	Y	N	N
General Call Flag (SSR[GCF])	Target detected an address match with the general call address.	Y	N	N
SMBus Alert Response Flag (SSR[SARF])	Target detected an address match with the SMBus alert address.	Y	N	N
Target Busy Flag (SSR[SBF])	LPI2C target is busy receiving an address byte or is transmitting or receiving data.	N	N	N
Bus Busy Flag (SSR[BBF])	LPI2C target is enabled and a Start condition is detected on I2C bus, but no Stop condition detected.	N	N	N

31.3.8.3 End-of-packet DMA transfer

End-of-packet functionality serves serial interfaces where the transfer size may not be known in advance and the data is pushed by an external device. Examples include UART receive, I2C Target mode, and SPI Target mode. End-of-packet processing is intended to ensure that data does not become stranded in either the receive FIFO or DMA receive buffer. Support for end-of-packet processing must be implemented in both the serial interfaces and DMA controller.

The condition that signals the end of packet differs for each serial interface but is processed by the serial peripheral and DMA in the same way. For example:

- UART end of packet is signaled by an idle line condition.
- I2C end of packet is signaled by Stop or repeated Start condition.
- SPI end of packet is signaled by PCS negation.

When the serial peripheral is configured to signal an end-of-packet condition to the DMA and the serial interface detects an end-of-packet condition, it asserts the DMA request for the receive FIFO regardless of the watermark configuration. For larger watermark configurations, this behavior ensures that the last few words of the transfer are first flushed from the receive FIFO.

The DMA then reads the contents of the receive FIFO, depending on the first word in the FIFO:

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

- If the receive FIFO is empty, the serial interface signals an end-of-packet condition to the DMA controller.
- If the receive FIFO is not empty, but the first word in the FIFO starts a new packet, data is not pulled from the receive FIFO. Also, the serial interface signals an end-of-packet condition to the DMA controller.
- If the receive FIFO is not empty, and the first word in the FIFO does not start a new packet, the DMA transfers data as normal.

Because the DMA may transfer multiple words per request, the end-of-packet condition persists until the DMA minor loop finishes and no additional data is pulled from the FIFO. The status flag that triggered the end-of-packet condition is cleared when the minor loop completes after the end of packet is signaled to the DMA controller.

When the DMA detects the end-of-packet condition, it writes all received words up to the end of packet into system memory. It also saves the destination address for the word after the last valid data. The DMA then terminates the channel as if the major loop completed, including final offsets and optional interrupts, channel linking, and scatter-gather. The final destination address can be saved to system memory or is available in the destination address register.

Because the DMA terminates the major loop, the receive FIFO is not serviced until the DMA is reconfigured. You can perform this reconfiguration through software or hardware (for example, channel linking or scatter-gather). Minimize this delay to avoid receiver FIFO overrun. Automatic DMA end-of-packet processing is not recommended when there are only a few words transferred between end-of-packet conditions. In this case, the DMA spends more time processing the end of packet than transferring the data. For example, to avoid an excessive number of idle conditions, increase the UART idle line length as needed.

31.4 External signals

Table 183. External signals

Signal	Name	Two-wire scheme	Four-wire scheme	Direction
SCL	LPI2C clock line	SCL	In Four-Wire mode, this pin is the SCL input pin.	Input or output
SDA	LPI2C data line	SDA	In Four-Wire mode, this pin is the SDA input pin.	Input or output
SCLS	Secondary I2C clock line	Not used	In Four-Wire mode, this pin is the SCLS output pin. If LPI2C controller/target are configured to use separate pins, then this pin is the LPI2C target SCL pin.	Input or output
SDAS	Secondary I2C data line	Not used	In Four-Wire mode, this pin is the SDAS output pin. If LPI2C controller/target are configured to use separate pins, then this pin is the LPI2C target SDA pin.	Input or output
HREQ	Host request	When configured for controller (input), if host request is asserted and the I2C bus is idle, it initiates a transfer. When configured for target (output), it asserts while valid data is present in Target Transmit Data (STDR). HREQ is an additional pin separate from the two-wire or four-wire scheme.		Input or output

Figure 132 shows the two-signal connection.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

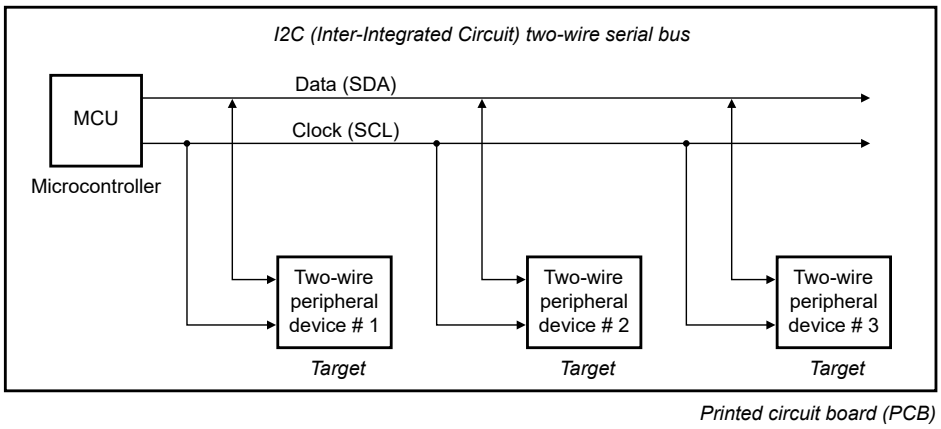


Figure 132. I²C two-wire serial bus

Figure 133 shows a possible four-signal connection.

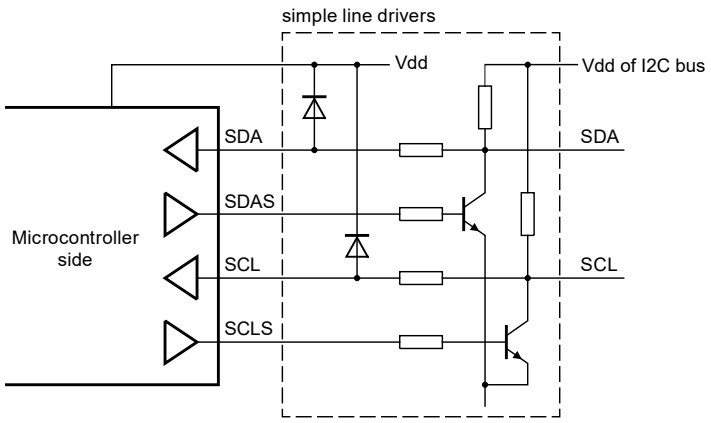


Figure 133. I²C four-wire serial bus

31.5 Initialization

To initialize the LPI2C controller:

1. Configure Controller Configuration 0 (MCFGR0)–Controller Configuration 3 (MCFGR3) as required by the application.
2. Configure Controller Clock Configuration 0 (MCCR0) and Controller Clock Configuration 1 (MCCR1) to satisfy the timing requirements of the I2C mode supported by the application.
3. Enable controller interrupts and DMA requests as required by the application.
4. Enable the LPI2C controller by writing 1 to MCR[MEN].

To initialize the LPI2C target:

1. Configure Target Address Match (SAMR) with the I2C address of the target location on the I2C bus.
2. Configure Target Configuration 0 (SCFGR0) and Target Configuration 1 (SCFGR1) as required by the application.
3. Configure Target Configuration 2 (SCFGR2) to satisfy the timing requirements of the I2C mode supported by the application.
4. Enable target interrupts and DMA requests as required by the application.
5. Enable the LPI2C target by writing 1 to SCR[SEN].

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

31.6 Application information

Configure the I2C timing parameters to meet the requirements of the I2C specification. This configuration depends on the supported mode and LPI2C functional clock frequency. When switching between two modes using different clock configuration registers (for example, Fast mode and HS mode), MCFGR1[PRESCALE] must remain constant between the modes.

Table 184. Example timing configurations

I2C mode	Clock frequency	Baud rate	PRESCALE	FILTSCL / FILTSDA	SETHOLD	CLKLO	CLKHI	DATAVD
Standard	8 MHz	100 kbit/s	0h	0h/0h	24h	28h	24h	02h
Standard	48 MHz	100 kbit/s	2h	1h/1h	37h	3Fh	37h	03h
Standard	60 MHz	100 kbit/s	2h	1h/1h	45h	50h	44h	04h
Fast	8 MHz	400 kbit/s	0h	0h/0h	04h	0Bh	05h	02h
Fast+	8 MHz	1 Mbit/s	0h	0h/0h	02h	03h	01h	01h
Fast	48 MHz	400 kbit/s	0h	1h/1h	1Dh	3Eh	35h	0Fh
Fast	48 MHz	400 kbit/s	2h	1h/1h	07h	11h	0Bh	03h
Fast+	48 MHz	1 Mbit/s	2h	1h/1h	03h	06h	04h	04h
HS	48 MHz	3.2 Mbit/s	0h	0h/0h	07h	08h	03h	01h
Fast	60 MHz	400 kbit/s	1h	2h/2h	11h	28h	1Fh	08h
Fast+	60 MHz	1 Mbit/s	1h	2h/2h	07h	0Fh	0Bh	01h
HS	60 MHz	3.33 Mbit/s	1h	0h/0h	04h	04h	02h	01h

31.7 Memory map and registers

31.7.1 LPI2C register descriptions

Writing to a read-only register or reading from a write-only register can cause bus errors. This module does not check whether programmed values in the registers are correct; you must ensure that valid programmed values are written to the registers.

31.7.1.1 LPI2C memory map

LPI2C0 base address: 4009_A000h

Offset	Register	Width (In bits)	Access	Reset value
0h	Version ID (VERID)	32	R	0103_0003h
4h	Parameter (PARAM)	32	R	0000_0202h
10h	Controller Control (MCR)	32	RW	0000_0000h
14h	Controller Status (MSR)	32	RW	0000_0001h
18h	Controller Interrupt Enable (MIER)	32	RW	0000_0000h
1Ch	Controller DMA Enable (MDER)	32	RW	0000_0000h
20h	Controller Configuration 0 (MCFGR0)	32	RW	0000_0000h

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Offset	Register	Width (In bits)	Access	Reset value
24h	Controller Configuration 1 (MCFGR1)	32	RW	0000_0000h
28h	Controller Configuration 2 (MCFGR2)	32	RW	0000_0000h
2Ch	Controller Configuration 3 (MCFGR3)	32	RW	0000_0000h
40h	Controller Data Match (MDMR)	32	RW	0000_0000h
48h	Controller Clock Configuration 0 (MCCR0)	32	RW	0000_0000h
50h	Controller Clock Configuration 1 (MCCR1)	32	RW	0000_0000h
58h	Controller FIFO Control (MFCR)	32	RW	0000_0000h
5Ch	Controller FIFO Status (MFSR)	32	R	0000_0000h
60h	Controller Transmit Data (MTDR)	32	W	0000_0000h
70h	Controller Receive Data (MRDR)	32	R	0000_4000h
78h	Controller Receive Data Read Only (MRDROR)	32	R	0000_4000h
110h	Target Control (SCR)	32	RW	0000_0000h
114h	Target Status (SSR)	32	RW	0000_0000h
118h	Target Interrupt Enable (SIER)	32	RW	0000_0000h
11Ch	Target DMA Enable (SDER)	32	RW	0000_0000h
120h	Target Configuration 0 (SCFGR0)	32	RW	0000_0000h
124h	Target Configuration 1 (SCFGR1)	32	RW	0000_0000h
128h	Target Configuration 2 (SCFGR2)	32	RW	0000_0000h
140h	Target Address Match (SAMR)	32	RW	0000_0000h
150h	Target Address Status (SASR)	32	R	0000_4000h
154h	Target Transmit ACK (STAR)	32	RW	0000_0000h
160h	Target Transmit Data (STDR)	32	W	0000_0000h
170h	Target Receive Data (SRDR)	32	R	0000_4000h
178h	Target Receive Data Read Only (SRDROR)	32	R	0000_4000h

31.7.1.2 Version ID (VERID)

Offset

Register	Offset
VERID	0h

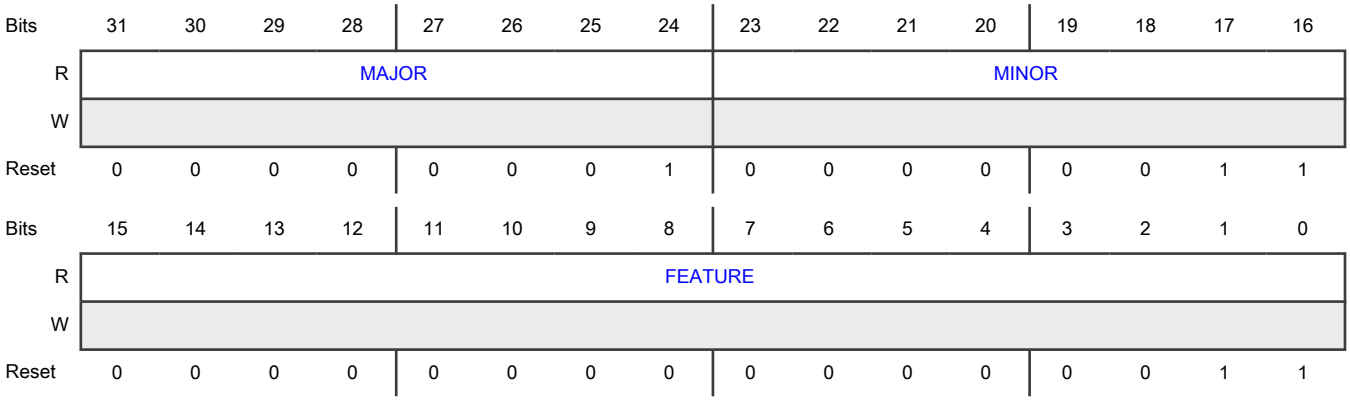
All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Function

Contains version numbers for the module design and feature set.

Diagram



Fields

Field	Function
31-24 MAJOR	Major Version Number Returns the major version number for the module design specification.
23-16 MINOR	Minor Version Number Returns the minor version number for the module design specification.
15-0 FEATURE	Feature Specification Number Returns the feature set number. 0000_0000_0000_0010b - Controller only, with standard feature set 0000_0000_0000_0011b - Controller and target, with standard feature set

31.7.1.3 Parameter (PARAM)

Offset

Register	Offset
PARAM	4h

Function

Contains parameter values implemented in the module.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0				MRXFIFO				0				MTXFIFO			
W																
Reset	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0

Fields

Field	Function
31-16 —	Reserved
15-12 —	Reserved
11-8 MRXFIFO	Controller Receive FIFO Size Configures the number of words in the controller receive FIFO to $2^{MRXFIFO}$.
7-4 —	Reserved
3-0 MTXFIFO	Controller Transmit FIFO Size Configures the number of words in the controller transmit FIFO to $2^{MTXFIFO}$.

31.7.1.4 Controller Control (MCR)

Offset

Register	Offset
MCR	10h

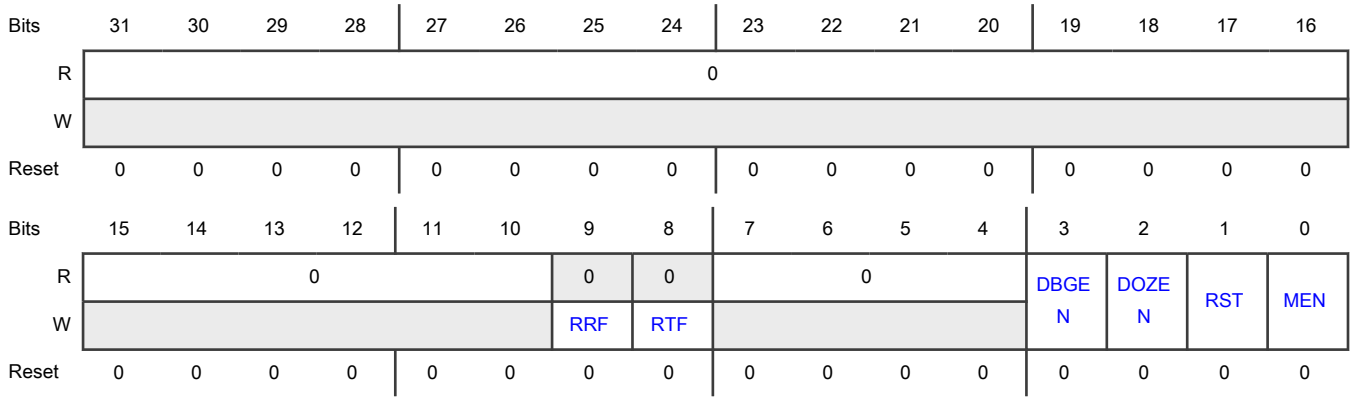
Function

Contains resets, debug enable, and other controller control settings.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram



Fields

Field	Function
31-10 —	Reserved
9 RRF	Reset Receive FIFO Resets the receive FIFO in Controller Receive Data (MRDR). 0b - No effect 1b - Reset receive FIFO
8 RTF	Reset Transmit FIFO Resets the transmit FIFO in Controller Transmit Data (MTDR). 0b - No effect 1b - Reset transmit FIFO
7-4 —	Reserved
3 DBGEN	Debug Enable Enables the controller in Debug mode. 0b - Disable 1b - Enable
2 DOZEN	Doze Mode Enable Enables the controller in Doze mode. 0b - Enable 1b - Disable
1	Software Reset

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
RST	Resets all internal controller logic and registers except Controller Control (MCR). This field remains 1 (enabled) until you write 0 to it. The reset takes effect immediately and remains asserted until negated by software. There is no minimum delay required before clearing the software reset. 0b - No effect 1b - Reset
0 MEN	Controller Enable Enables the controller logic. 0b - Disable 1b - Enable

31.7.1.5 Controller Status (MSR)

Offset

Register	Offset
MSR	14h

Function

Contains status flags for transmit and receive data, for start and stop conditions, and for bus and controller busy or idle status.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0						BBF	MBF	0							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	STF	DMF	PLTF	FEF	ALF	NDF	SDF	EPF	0						RDF	TDF
W	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C								
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Fields

Field	Function
31-26 —	Reserved

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
25 BBF	Bus Busy Flag Specifies whether the I2C bus is busy. 0b - Idle 1b - Busy
24 MBF	Controller Busy Flag Specifies whether the I2C controller is busy. 0b - Idle 1b - Busy
23-16 —	Reserved
15 STF	Start Flag Specifies whether a Start condition is detected on the bus when the bus is idle. When MCFGR1[STARTCFG] is 0, this field becomes 1 only if the LPI2C controller is also idle. When MCFGR1[STARTCFG] is 1, STF becomes 1 for any Start condition when the I2C bus is idle. **NOTE** This field behaves differently for register reads and writes. When reading 0b - Start condition not detected 1b - Start condition detected When writing 0b - No effect 1b - Clear the flag
14 DMF	Data Match Flag Indicates whether the received data matches MDMR[MATCH0] or MDMR[MATCH1] (as configured by MCFGR1[MATCFG]). Received data discarded due to MTDR[CMD] does not cause this flag to set. **NOTE** This field behaves differently for register reads and writes. When reading 0b - Matching data not received 1b - Matching data received When writing 0b - No effect 1b - Clear the flag

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
13 PLTF	Pin Low Timeout Flag Indicates whether pin low timeout has occurred. Sets when the SCL or SDA input is low for more than the number of PINLOW cycles defined by MCFGR3[PINLOW], even when the LPI2C controller is idle. You must resolve the pin low condition via software. PLTF cannot be cleared as long as the pin low timeout continues. Before LPI2C can initiate a Start condition, you must clear this flag. See MCFGR1[TIMECFG] for the SCL and/or SDA timeout settings. **NOTE** This field behaves differently for register reads and writes. When reading 0b - Pin low timeout did not occur 1b - Pin low timeout occurred When writing 0b - No effect 1b - Clear the flag
12 FEF	FIFO Error Flag Detects the LPI2C controller's attempt to send or receive data without first generating a (repeated) Start condition. This error can occur when the transmit FIFO underflows when MCFGR1[AUTOSTOP] = 1. When this flag is set, the LPI2C controller sends a Stop condition (if busy). The controller does not initiate a new Start condition until the flag is cleared. **NOTE** This field behaves differently for register reads and writes. When reading 0b - No FIFO error 1b - FIFO error When writing 0b - No effect 1b - Clear the flag
11 ALF	Arbitration Lost Flag Indicates whether arbitration is lost. Either of these conditions sets this flag: • The LPI2C controller transmits a logic 1 and detects a logic 0 on the I2C bus. • The LPI2C controller detects a Start or Stop condition when the LPI2C controller is transmitting data. When ALF is set, the LPI2C controller releases the I2C bus (goes idle), and the LPI2C controller does not initiate a new Start condition until the ALF is cleared.

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	NOTE This field behaves differently for register reads and writes. When reading 0b - Controller did not lose arbitration 1b - Controller lost arbitration When writing 0b - No effect 1b - Clear the flag
10 NDF	NACK Detect Flag Indicates whether an unexpected NACK has been detected. This flag is set when the LPI2C controller detects a NACK it was not expecting when transmitting an address or data. When set, the controller does not initiate a new Start condition until this flag is cleared. If a NACK is expected for a given address (as configured by the command word), this flag is set if a NACK is not generated. When this flag is set, the LPI2C controller automatically transmits a Stop condition if MCFGR1[AUTOSTOP] = 1, or if the transmit FIFO is not empty. **NOTE** This field behaves differently for register reads and writes. When reading 0b - No unexpected NACK detected 1b - Unexpected NACK detected When writing 0b - No effect 1b - Clear the flag
9 SDF	Stop Detect Flag Indicates whether the LPI2C controller has generated a Stop condition. When MCFGR1[STOPCFG] = 0, this flag is set for any Stop condition. When MCFGR1[STOPCFG] = 1, this flag is set only when the LPI2C controller is also idle (transmit FIFO is empty). **NOTE** This field behaves differently for register reads and writes. When reading 0b - No Stop condition generated 1b - Stop condition generated When writing 0b - No effect

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	1b - Clear the flag
8 EPF	End Packet Flag Indicates whether the LPI2C controller has generated a repeated Start condition or a Stop condition. When the controller first generates a Start condition, this flag is not set. **NOTE** This field behaves differently for register reads and writes. When reading 0b - No Stop or repeated Start generated 1b - Stop or repeated Start generated When writing 0b - No effect 1b - Clear the flag
7-2 —	Reserved
1 RDF	Receive Data Flag Indicates whether the receive data is ready. This flag is set when the number of words in the receive FIFO is greater than MFCR[RXWATER]. 0b - Receive data not ready 1b - Receive data ready
0 TDF	Transmit Data Flag Indicates whether transmit data is requested. This flag is set when the number of words in the transmit FIFO is equal or less than MFCR[TXWATER]. 0b - Transmit data not requested 1b - Transmit data requested

31.7.1.6 Controller Interrupt Enable (MIER)

Offset

Register	Offset
MIER	18h

Function

Contains enables for:

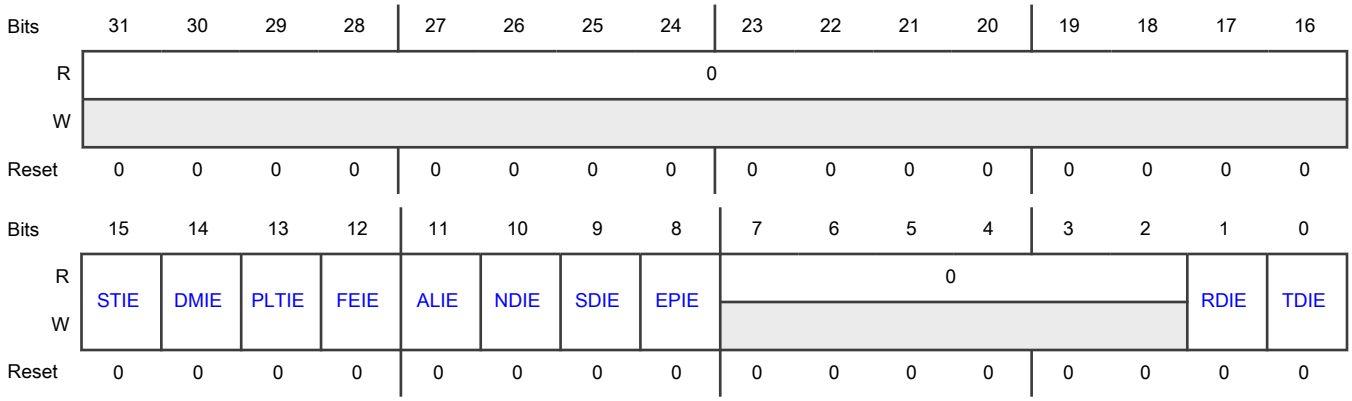
- Transmit and receive data interrupts

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

- Start, Stop, and NACK detection interrupts
- DMA interrupts

Diagram



Fields

Field	Function
31-16 —	Reserved
15 STIE	Start Interrupt Enable Enables interrupt for Start condition. 0b - Disable 1b - Enable
14 DMIE	Data Match Interrupt Enable Enables interrupt for data match. 0b - Disable 1b - Enable
13 PLTIE	Pin Low Timeout Interrupt Enable Enables interrupt for pin-low timeout. 0b - Disable 1b - Enable
12 FEIE	FIFO Error Interrupt Enable Enables interrupt for FIFO error. 0b - Disable 1b - Enable
11	Arbitration Lost Interrupt Enable

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
ALIE	Enables interrupt for arbitration lost. 0b - Disable 1b - Enable
10 NDIE	NACK Detect Interrupt Enable Enables interrupt for NACK detection. 0b - Disable 1b - Enable
9 SDIE	Stop Detect Interrupt Enable Enables interrupt for Stop detection. 0b - Disable 1b - Enable
8 EPIE	End Packet Interrupt Enable Enables interrupt for end packet. 0b - Disable 1b - Enable
7-2 —	Reserved
1 RDIE	Receive Data Interrupt Enable Enables interrupt for receive data. 0b - Disable 1b - Enable
0 TDIE	Transmit Data Interrupt Enable Enables interrupt for transmit data. 0b - Disable 1b - Enable

31.7.1.7 Controller DMA Enable (MDER)

Offset

Register	Offset
MDER	1Ch

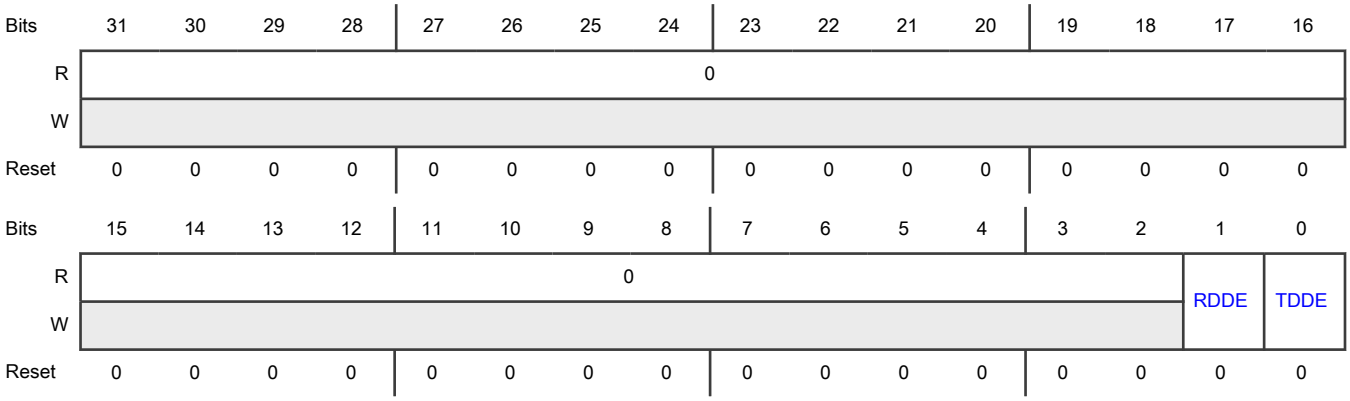
All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Function

Contains DMA transmit, request, and receive enables.

Diagram



Fields

Field	Function
31-2 —	Reserved
1 RDDE	Receive Data DMA Enable Enables DMA receive data. 0b - Disable 1b - Enable
0 TDDE	Transmit Data DMA Enable Enables DMA transmit data. 0b - Disable 1b - Enable

31.7.1.8 Controller Configuration 0 (MCFGR0)

Offset

Register	Offset
MCFGR0	20h

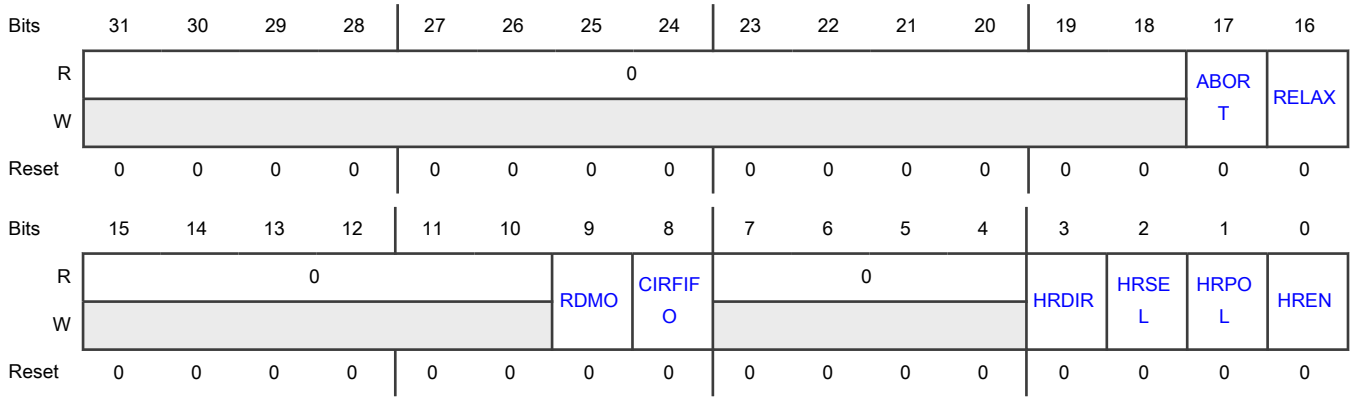
Function

Contains host settings and other receive and transfer settings.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram



Fields

Field	Function
31-18 —	Reserved
17 ABORT	Abort Transfer Determines whether the LPI2C controller completes the existing byte and then sends a Stop condition. A new transfer does not start while this field is 1. 0b - Normal transfer 1b - Abort existing transfer and do not start a new one
16 RELAX	Relaxed Mode Specifies the type of transfer. If this field is 1, the LPI2C controller relaxes several transfer restrictions. Transfers start when the bus is busy and FIFO commands are not checked for errors. You can use this mode to attempt recovery of a stuck I2C target by toggling the SCL pin. 0b - Normal transfer 1b - Relaxed transfer
15-10 —	Reserved
9 RDMO	Receive Data Match Only Determines whether all received data that does not set MSR[DMF] is discarded. After MSR[DMF] is set, the RDMO configuration is ignored. When disabling RDMO, write 0 to this field before writing 0 to MSR[DMF] to ensure that no receive data is lost. 0b - Received data is stored in the receive FIFO 1b - Received data is discarded unless MSR[DMF] is set
8 CIRFIFO	Circular FIFO Enable Enables the transmit FIFO read pointer to be saved to a temporary register. The transmit FIFO empties as normal. After the LPI2C controller is idle and the transmit FIFO is empty, the read pointer value is

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	restored from the temporary register. This setting causes the contents of the transmit FIFO to be cycled through repeatedly. If MCFGR1[AUTOSTOP] is 1, then a Stop condition is sent whenever the transmit FIFO is empty and the read pointer is restored. 0b - Disable 1b - Enable
7-4 —	Reserved
3 HRDIR	Host Request Direction Configures the direction of the HREQ pin. When this field is 1, the LPI2C target drives the HREQ output pin and the pin becomes 1 when there is data in the target transmit data register. For proper operation when this field is 1, write 1 to SCFGR1[TXCFG]. 0b - HREQ pin is input (for LPI2C controller) 1b - HREQ pin is output (for LPI2C target)
2 HRSEL	Host Request Select Selects the source of the host request input. When host request is enabled, this field must not change. 0b - Host request input is pin HREQ 1b - Host request input is input trigger
1 HRPOL	Host Request Polarity Configures the polarity of the host request input. When host request is enabled, this field must not change. HRPOL sets the polarity for both the HREQ pin and the input trigger. • When HRPOL=0, the polarity is configured for active low, so host request is asserted if the HREQ pin or input trigger are logic 0. • When HRPOL=1, the polarity is configured for active high, so host request is asserted if the HREQ pin or input trigger are logic 1. 0b - Active low 1b - Active high
0 HREN	Host Request Enable Enables host request. When enabled, the LPI2C controller only initiates a Start condition if the host request input is asserted and the bus is idle. A repeated Start condition is not affected by the host request. 0b - Disable 1b - Enable

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

31.7.1.9 Controller Configuration 1 (MCFGR1)

Offset

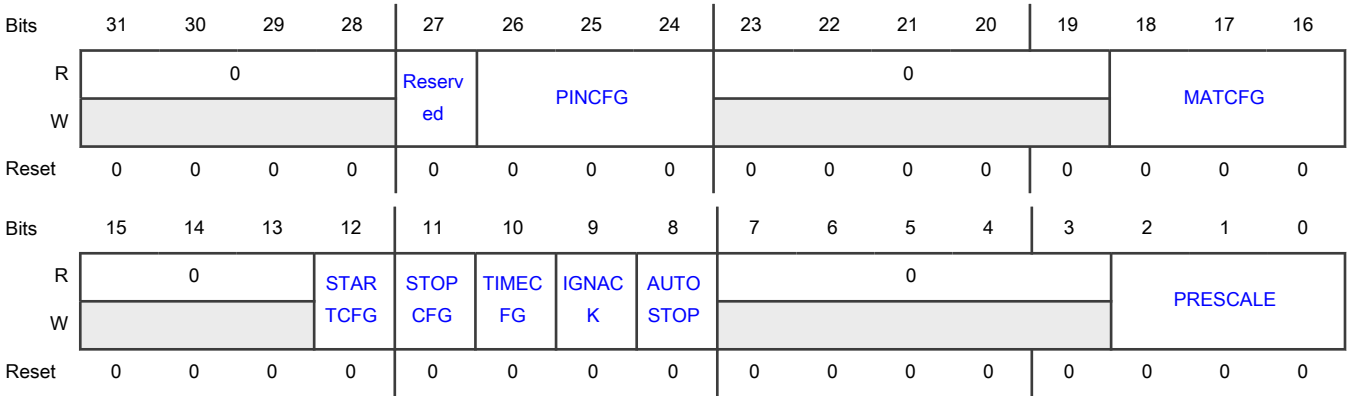
Register	Offset
MCFGR1	24h

Function

Contains controls for pin configuration, clock prescaler, and various other control settings.

Write to this register only when the I2C controller is disabled.

Diagram



Fields

Field	Function
31-28 —	Reserved
27 —	Reserved
26-24 PINCFG	Pin Configuration Configures the pin mode for LPI2C. 000b - Two-pin open drain mode. SCL/SDA pins: Bidirectional open drain for controller and target. SCLS/SDAS pins: Not used. 001b - Two-pin output only mode (Ultra-Fast mode). SCL/SDA pins: Output-only (Ultra-Fast mode) open drain for controller and target. SCLS/SDAS pins: Not used. 010b - Two-pin push-pull mode. SCL/SDA pins: Bidirectional push-pull for controller and target. SCLS/SDAS pins: Not used. 011b - Four-pin push-pull mode. SCL/SDA pins: Input only for controller and target. SCLS/SDAS pins: Output-only push-pull for controller and target.

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	100b - Two-pin open-drain mode with separate LPI2C target. SCL/SDA pins: Bidirectional open drain for controller. SCLS/SDAS pins: Bidirectional open drain for target. 101b - Two-pin output only mode (Ultra-Fast mode) with separate LPI2C target. SCL/SDA pins: Output-only (Ultra-Fast mode) open drain for controller. SCLS/SDAS pins: Output-only open drain for target. 110b - Two-pin push-pull mode with separate LPI2C target. SCL/SDA pins: Bidirectional push-pull for controller. SCLS/SDAS pins: Bidirectional push-pull for target. 111b - Four-pin push-pull mode (inverted outputs). SCL/SDA pins: Input only for controller and target. SCLS/SDAS pins: Inverted output-only push-pull for controller and target.
23-19 —	Reserved
18-16 MATCFG	Match Configuration Configures the condition that sets MSR[DMF]. See Controller Data Match (MDMR). 000b - Match is disabled 001b - Reserved 010b - Match is enabled: first data word equals MDMR[MATCH0] OR MDMR[MATCH1] 011b - Match is enabled: any data word equals MDMR[MATCH0] OR MDMR[MATCH1] 100b - Match is enabled: (first data word equals MDMR[MATCH0]) AND (second data word equals MDMR[MATCH1) 101b - Match is enabled: (any data word equals MDMR[MATCH0]) AND (next data word equals MDMR[MATCH1) 110b - Match is enabled: (first data word AND MDMR[MATCH1]) equals (MDMR[MATCH0] AND MDMR[MATCH1]) 111b - Match is enabled: (any data word AND MDMR[MATCH1]) equals (MDMR[MATCH0] AND MDMR[MATCH1])
15-13 —	Reserved
12 STARTCFG	Start Configuration Configures the conditions that set MSR[STF] when a Start condition occurs. When this field is 0, MSR[STF] is set on a Start condition when both the I2C bus and LPI2C are idle. In other words, any nonrepeated Start condition is initiated by any controller on the bus except the LPI2C controller. When this field is 1, MSR[STF] is set on a Start condition when the I2C bus is idle. In other words, any nonrepeated Start condition is initiated by any controller on the bus including the LPI2C controller. 0b - Sets when both I2C bus and LPI2C controller are idle 1b - Sets when I2C bus is idle

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
11 STOPCFG	Stop Configuration Configures the conditions that set MSR[SDF]. When this field is 0, any Stop condition generated by the LPI2C controller sets SDF. When this field is 1, the last Stop condition before the LPI2C controller is idle sets SDF. In this case, the transmit FIFO is empty at the time of the Stop condition. 0b - Any Stop condition 1b - Last Stop condition
10 TIMECFG	Timeout Configuration Configures which signals must be low for longer than the configured timeout to set MSR[PLTF]. When this field is 0, MSR[PLTF] is set when SCL is low for longer than the configured timeout. 0b - SCL 1b - SCL or SDA
9 IGNACK	Ignore NACK Determines whether the LPI2C controller ignores a received NACK and treats it as an ACK. In this case, MSR[NDF] is never set. This field must be 1 in Ultra-Fast mode. 0b - No effect 1b - Treat a received NACK as an ACK
8 AUTOSTOP	Automatic Stop Generation Determines whether a Stop condition is generated when the LPI2C controller is busy and the transmit FIFO is empty. A Stop condition can also be generated using a transmit FIFO command. When this field is 1, a Stop condition is automatically generated when the transmit FIFO is empty and the LPI2C controller is busy. 0b - No effect 1b - Stop automatically generated
7-3 —	Reserved
2-0 PRESCALE	Prescaler Configures the clock prescaler used for all LPI2C controller logic except the digital glitch filters. 000b - Divide by 1 001b - Divide by 2 010b - Divide by 4 011b - Divide by 8 100b - Divide by 16

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	101b - Divide by 32 110b - Divide by 64 111b - Divide by 128

31.7.1.10 Controller Configuration 2 (MCFGR2)

Offset

Register	Offset
MCFGR2	28h

Function

Contains the configuration for the bus idle timeout and glitch filters for SDA and SCL.

Write to this register only when the I2C controller is disabled.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0				FILTSDA				0				FILTSCL			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0				BUSIDLE											
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-28 —	Reserved
27-24 FILTSDA	Glitch Filter SDA Configures the I2C controller digital glitch filters for the SDA input. The latency through the glitch filter is equal to the number of cycles defined by this field. The value of this field must be less than the minimum SCL low or high period.

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	Glitches equal to or less than the number of cycles defined by this field are filtered out and ignored. Writing 0 to this field disables the glitch filter. MCFGR1[PRESCALE] does not affect the glitch filter cycle count. It is automatically bypassed in HS mode.
23-20 —	Reserved
19-16 FILTSCL	Glitch Filter SCL Configures the I2C controller digital glitch filters for SCL input. The latency through the glitch filter is equal to the number of cycles defined by this field. The value of this field must be less than the minimum SCL low or high period. Glitches equal to or less than the number of cycles defined by this field are filtered out and ignored. These cycles are based on the functional clock. Writing 0 to this field disables the glitch filter. MCFGR1[PRESCALE] does not affect the glitch filter cycle count. It is automatically bypassed in HS mode.
15-12 —	Reserved
11-0 BUSIDLE	Bus Idle Timeout Configures the bus idle timeout period, in clock cycles. If both SCL and SDA are high for longer than the number of cycles defined by this field, the I2C bus is assumed to be idle and the controller can generate a Start condition. Writing 0 to this field disables the bus idle timeout.

31.7.1.11 Controller Configuration 3 (MCFGR3)

Offset

Register	Offset
MCFGR3	2Ch

Function

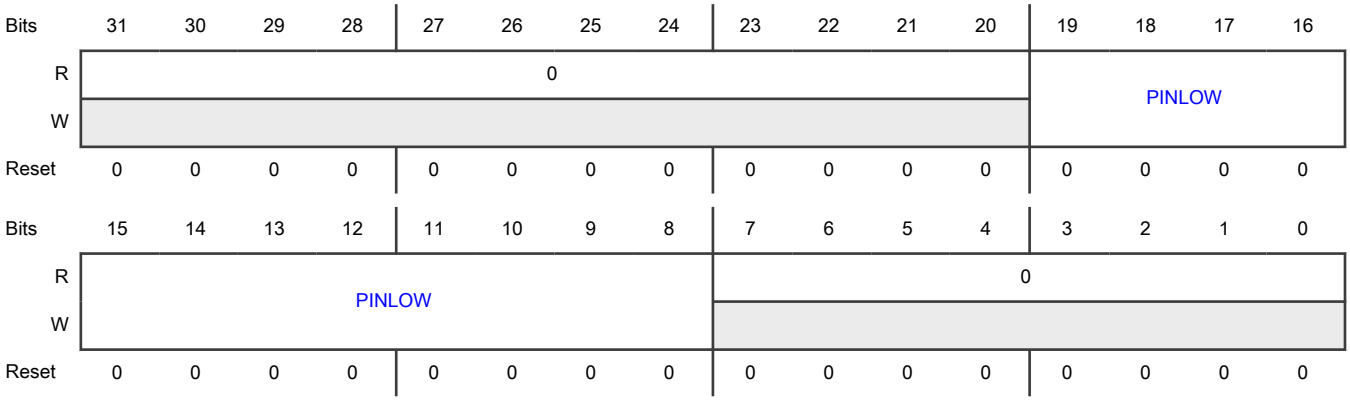
Configures the threshold value for the pin low timeout flag.

Write to this register only when the I2C controller is disabled.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram



Fields

Field	Function
31-20 —	Reserved
19-8 PINLOW	Pin Low Timeout Configures the threshold value, in clock cycles, that sets MSR[PLTF]. If SCL or SDA (selected by MCFGR1[TIMECFG]) is low for longer than (PINLOW × 256) cycles, MSR[PLTF] is set. When this field is 0, the pin low timeout feature is disabled.
7-0 —	Reserved

31.7.1.12 Controller Data Match (MDMR)

Offset

Register	Offset
MDMR	40h

Function

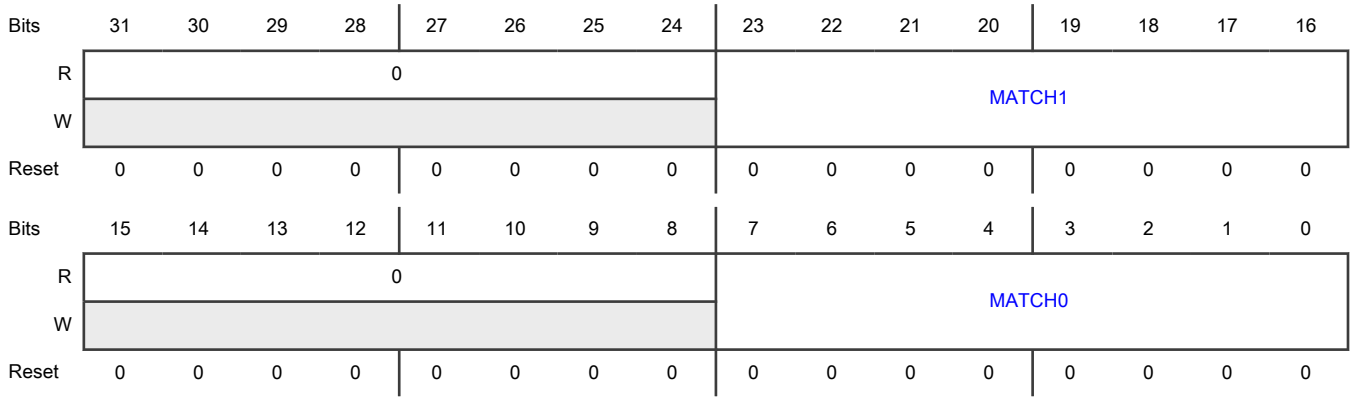
Contains data match values.

Write to this register only when the I2C controller is disabled or idle.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram



Fields

Field	Function
31-24 —	Reserved
23-16 MATCH1	Match 1 Value Specifies match 1 value that is compared to the received data when receive data match is enabled.
15-8 —	Reserved
7-0 MATCH0	Match 0 Value Specifies match 0 value that is compared to the received data when receive data match is enabled.

31.7.1.13 Controller Clock Configuration 0 (MCCR0)

Offset

Register	Offset
MCCR0	48h

Function

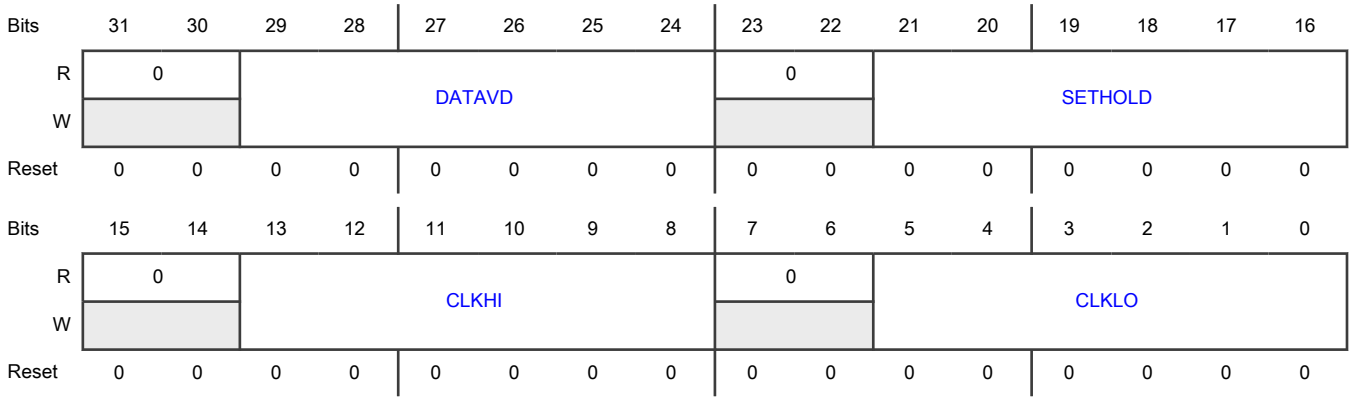
Configures various clock controls.

You cannot make changes to this register when the I2C controller is enabled and is used for standard, fast, fast-mode plus, and ultra-fast transfers.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram



Fields

Field	Function
31-30 —	Reserved
29-24 DATAVD	Data Valid Delay Specifies the minimum number of cycles (minus one) used as the data hold time for SDA. This value must be less than the minimum SCL low period.
23-22 —	Reserved
21-16 SETHOLD	Setup Hold Delay Specifies the minimum number of cycles (minus one) used by the controller for these conditions: • Hold time for a Start • Setup and hold time for a repeated Start • Setup time for a Stop The setup time is extended by the time it takes to detect a rising edge on the external SCL pin. Ignoring any additional board delay due to external loading, this time is equal to (2 + FILTSCL) ÷ 2^PRESCALE cycles.
15-14 —	Reserved
13-8 CLKHI	Clock High Period Specifies the minimum number of cycles (minus one) that the controller drives the SCL clock high. The SCL high time is extended by the time needed to detect a rising edge on the external SCL pin. Ignoring any additional board delay due to external loading, this time is equal to (2 + FILTSCL) ÷ 2^PRESCALE cycles.
7-6 —	Reserved

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
5-0 CLKLO	Clock Low Period Specifies the minimum number of cycles (minus one) that the controller drives the SCL clock low. This value is also used for the minimum bus free time between a Stop and a Start condition. This period is extended by the time needed to detect a rising edge on the external SCL pin. Ignoring any additional board delay due to external loading, this time is equal to (2 + FILTSCL) ÷ 2^PRESCALE cycles.

31.7.1.14 Controller Clock Configuration 1 (MCCR1)

Offset

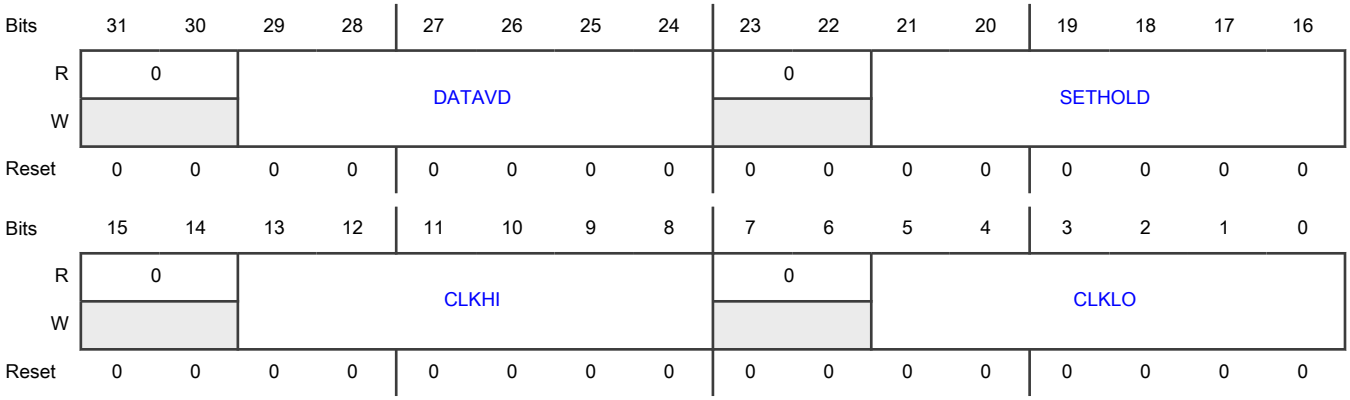
Register	Offset
MCCR1	50h

Function

Configures various clock controls.

You cannot makes changes to this register when the I2C controller is enabled and is used for HS mode transfers. The separate clock configuration for HS mode allows arbitration to take place in Fast mode (with timing configured by Controller Clock Configuration 0 (MCCR0)), before switching to HS mode (with timing configured by MCCR1).

Diagram



Fields

Field	Function
31-30 —	Reserved
29-24 DATAVD	Data Valid Delay

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	Specifies the minimum number of cycles (minus one) used as the data hold time for SDA. This value must be less than the minimum SCL low period.
23-22 —	Reserved
21-16 SETHOLD	Setup Hold Delay Specifies the minimum number of cycles (minus one) used by the controller for these conditions: • Hold time for a Start condition • Setup and hold time for a repeated Start condition • Setup time for a Stop condition The setup time is extended by the time needed to detect a rising edge on the external SCL pin. Ignoring any additional board delay due to external loading, this time is equal to (2 + FILTSCL) ÷ 2^PRESCALE cycles.
15-14 —	Reserved
13-8 CLKHI	Clock High Period Specifies the minimum number of cycles (minus one) that the controller drives the SCL clock high. The SCL high time is extended by the time needed to detect a rising edge on the external SCL pin. Ignoring any additional board delay due to external loading, this time is equal to (2 + FILTSCL) ÷ 2^PRESCALE cycles.
7-6 —	Reserved
5-0 CLKLO	Clock Low Period Specifies the minimum number of cycles (minus one) that the controller drives the SCL clock low. This value is also used for the minimum bus free time between a Stop and a Start condition. This period is extended by the time needed to detect a rising edge on the external SCL pin. Ignoring any additional board delay due to external loading, this time is equal to (2 + FILTSCL) ÷ 2^PRESCALE cycles.

31.7.1.15 Controller FIFO Control (MFCR)

Offset

Register	Offset
MFCR	58h

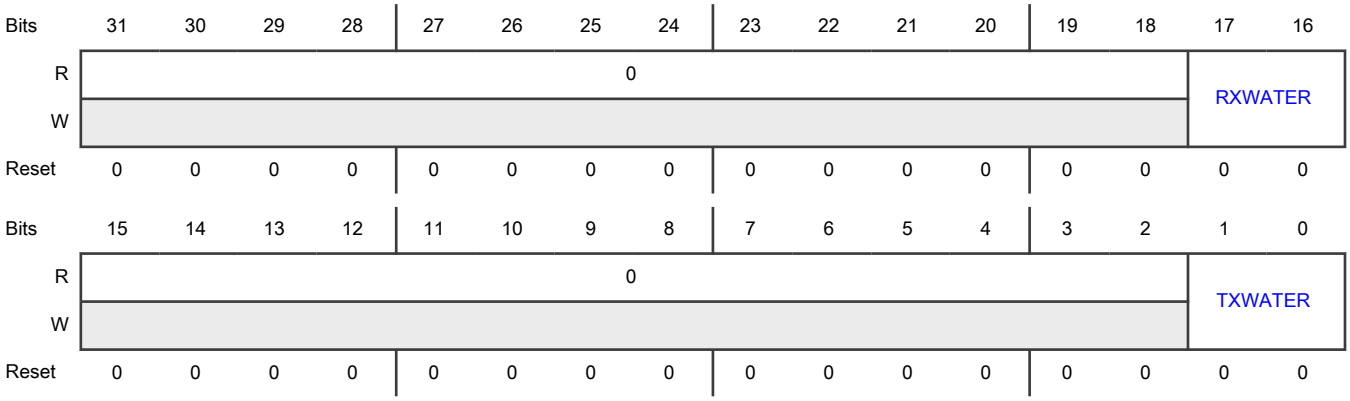
Function

Controls the receive and transmit FIFO watermark values.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram



Fields

Field	Function
31-18 —	Reserved
17-16 RXWATER	Receive FIFO Watermark Determines the watermark for setting MSR[RDF]. That flag is set when the number of words in the receive FIFO is greater than the value of this field. Writing a value equal to or greater than the FIFO size truncates the value.
15-2 —	Reserved
1-0 TXWATER	Transmit FIFO Watermark Determines the watermark for setting MSR[TDF]. That flag is set when the number of words in the transmit FIFO is equal or less than the value of this field. Writing a value equal to or greater than the FIFO size truncates the value.

31.7.1.16 Controller FIFO Status (MFSR)

Offset

Register	Offset
MFSR	5Ch

Function

Specifies the number of words in the transmit and receive FIFOs.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0													RXCOUNT		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0													TXCOUNT		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-19 —	Reserved
18-16 RXCOUNT	Receive FIFO Count Specifies the number of words in the receive FIFO.
15-3 —	Reserved
2-0 TXCOUNT	Transmit FIFO Count Specifies the number of words in the transmit FIFO.

31.7.1.17 Controller Transmit Data (MTDR)

Offset

Register	Offset
MTDR	60h

Function

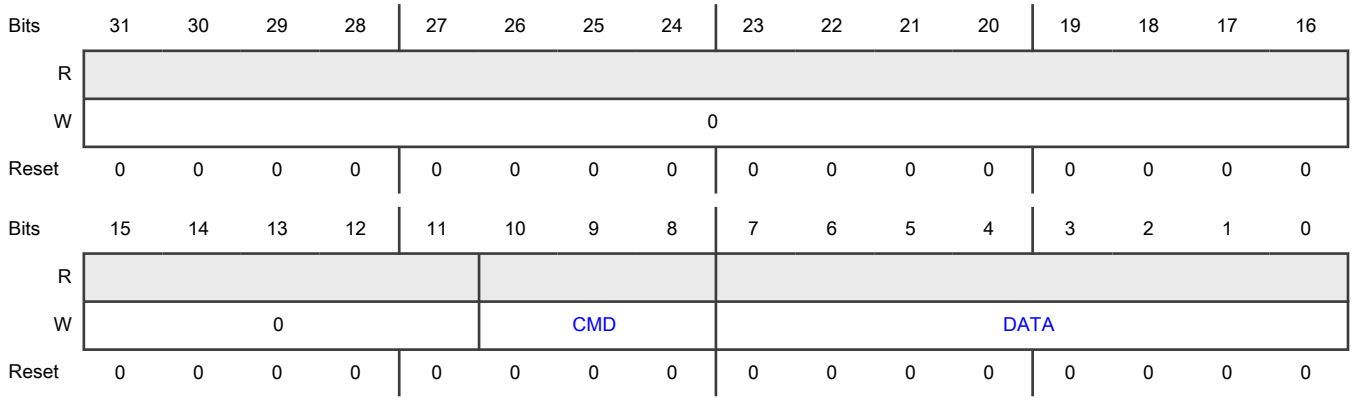
Configures transmit data:

- An 8-bit write to MTDR[CMD] is ignored and does not increment the FIFO write pointer.
- An 8-bit write to MTDR[DATA] zero-extends the value of MTDR[CMD] and increments the FIFO write pointer.
- A 16-bit or 32-bit write operation writes to both MTDR[CMD] and MTDR[DATA] and increments the FIFO write pointer.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram



Fields

Field	Function
31-11 —	Reserved
10-8 CMD	Command Data Selects command transmitted by controller. 000b - Transmit the value in DATA[7:0] 001b - Receive (DATA[7:0] + 1) bytes. DATA[7:0] is used as a byte counter. Receive that many bytes and check each for a data match (if configured) before storing the received data in the receive FIFO. 010b - Generate Stop condition on I2C bus 011b - Receive and discard (DATA[7:0] + 1) bytes. DATA[7:0] is used as a byte counter. Receive that many bytes but do not check for a data match or store those bytes in the receive FIFO. 100b - Generate (repeated) Start on the I2C bus and transmit the address in DATA[7:0] 101b - Generate (repeated) Start on the I2C bus and transmit the address in DATA[7:0] (this transfer expects a NACK to be returned) 110b - Generate (repeated) Start on the I2C bus and transmit the address in DATA[7:0] using HS mode 111b - Generate (repeated) Start on the I2C bus and transmit the address in DATA[7:0] using HS mode (this transfer expects a NACK to be returned)
7-0 DATA	Transmit Data Contains data used by the commands listed in MTDR[CMD]. Performing an 8-bit write to this field zero-extends the value of MTDR[CMD].

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

31.7.1.18 Controller Receive Data (MRDR)

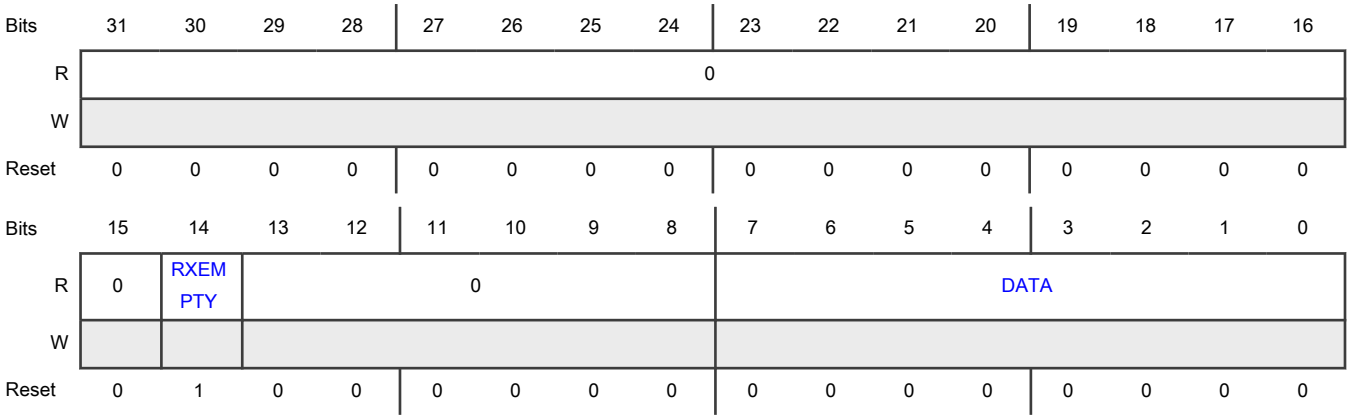
Offset

Register	Offset
MRDR	70h

Function

Contains the status of the receive FIFO and the data received by the I2C controller that has not been discarded.

Diagram



Fields

Field	Function
31-15 —	Reserved
14 RXEMPTY	Receive Empty Indicates whether the controller receive data FIFO is empty. 0b - Not empty 1b - Empty
13-8 —	Reserved
7-0 DATA	Receive Data Contains data received by the I2C controller that has not been discarded. Received data can be discarded due to the command in MTDR[CMD], or the controller can be configured to discard nonmatching data.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

31.7.1.19 Controller Receive Data Read Only (MRDROR)

Offset

Register	Offset
MRDROR	78h

Function

Contains the status of the receive FIFO and returns the data received by the I2C controller, but does not pull the data from the FIFO.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	RXEMPTY	0						DATA							
W																
Reset	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-15 —	Reserved
14 RXEMPTY	RX Empty Indicates whether the receive data FIFO is empty. 0b - Not empty 1b - Empty
13-8 —	Reserved
7-0 DATA	Receive Data

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

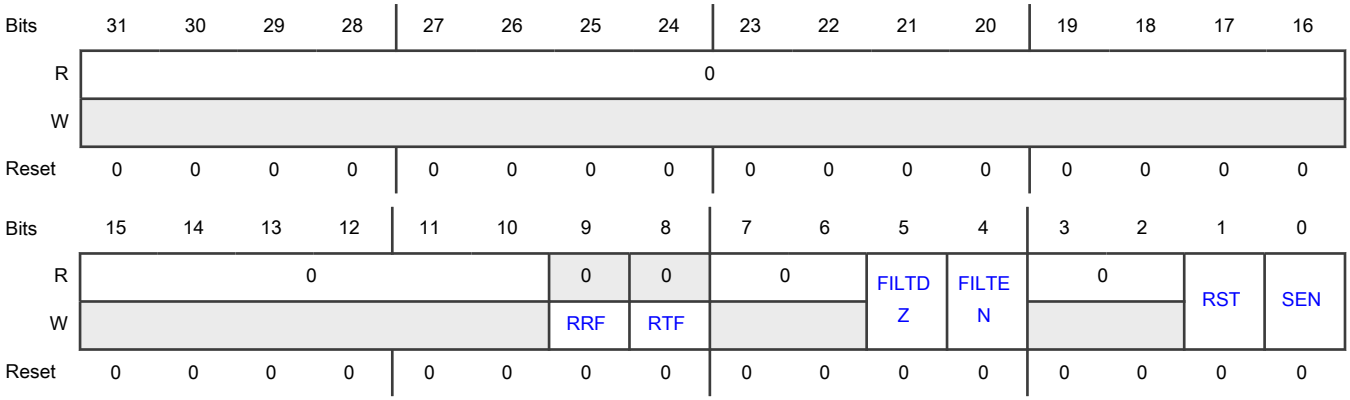
31.7.1.20 Target Control (SCR)

Offset

Register	Offset
SCR	110h

Function
Contains resets and other target control settings.

Diagram



Fields

Field	Function
31-10 —	Reserved
9 RRF	Reset Receive FIFO Empties the receive FIFO in Target Receive Data (SRDR). 0b - No effect 1b - SRDR is now empty
8 RTF	Reset Transmit FIFO Empties the transmit FIFO in Target Transmit Data (STDR). 0b - No effect 1b - STDR is now empty
7-6 —	Reserved
5	Filter Doze Enable

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
FILTDZ	Enables filter in Doze mode. Update this field only when the I2C target is disabled. 0b - Enable 1b - Disable
4 FILTEN	Filter Enable Enables digital filter and output delay counter for target mode. Update this field only when the I2C target is disabled. 0b - Disable 1b - Enable
3-2 —	Reserved
1 RST	Software Reset Resets target mode logic. The reset takes effect immediately. The value of this field remains 1 until you write 0 to it. There is no minimum delay required before clearing the software reset. 0b - Not reset 1b - Reset
0 SEN	Target Enable Enables I2C Target mode. 0b - Disable 1b - Enable

31.7.1.21 Target Status (SSR)

Offset

Register	Offset
SSR	114h

Function

Contains status flags for transmit and receive data, for error conditions, and for bus and target busy or idle status.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0						BBF	SBF	0							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SARF	GCF	AM1F	AM0F	FEF	BEF	SDF	RSF	0				TAF	AVF	RDF	TDF
W					W1C	W1C	W1C	W1C								
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-26 —	Reserved
25 BBF	Bus Busy Flag Indicates whether an I2C bus is idle or busy. 0b - Idle 1b - Busy
24 SBF	Target Busy Flag Indicates whether an I2C target is idle or busy. 0b - Idle 1b - Busy
23-16 —	Reserved
15 SARF	SMBus Alert Response Flag Indicates whether an SMBus alert response has been detected. You can clear this flag by reading Target Address Status (SASR). This flag cannot generate an asynchronous wakeup. 0b - Disabled or not detected 1b - Enabled and detected
14 GCF	General Call Flag Indicates whether a target has detected the general call address. You can clear this flag by reading Target Address Status (SASR). This flag cannot generate an asynchronous wakeup.

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	0b - General call address disabled or not detected 1b - General call address detected
13 AM1F	Address Match 1 Flag Indicates whether the received address matches the value in ADDR1, or it falls within the ADDR0 to ADDR1 range as configured by SCFGR1[ADDRCFG]. This flag is cleared by reading Target Address Status (SASR). This flag cannot generate an asynchronous wakeup. 0b - Matching address not received 1b - Matching address received
12 AM0F	Address Match 0 Flag Indicates whether the received address matches the ADDR0 field, as configured by SCFGR1[ADDRCFG]. This flag is cleared by reading Target Address Status (SASR). This flag cannot generate an asynchronous wakeup. 0b - ADDR0 matching address not received 1b - ADDR0 matching address received
11 FEF	FIFO Error Flag Indicates whether there is a FIFO error. This flag can only be set when clock stretching is disabled. **NOTE** This field behaves differently for register reads and writes. When reading 0b - No FIFO error 1b - FIFO error When writing 0b - No effect 1b - Clear the flag
10 BEF	Bit Error Flag Indicates whether the LPI2C target has transmitted a logic 1 and detects a logic 0 on the I2C bus. The target ignores the rest of the transfer until the next (repeated) Start condition. **NOTE** This field behaves differently for register reads and writes. When reading 0b - No bit error occurred

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	1b - Bit error occurred When writing 0b - No effect 1b - Clear the flag
9 SDF	Stop Detect Flag Indicates whether the LPI2C target detects a Stop condition, and if the LPI2C target matched the last address byte. When SCFGR1[SDCFG] = 1, this flag is set on any Stop condition. **NOTE** This field behaves differently for register reads and writes. When reading 0b - No Stop detected 1b - Stop detected When writing 0b - No effect 1b - Clear the flag
8 RSF	Repeated Start Flag Indicates whether the LPI2C target detects a repeated Start condition and if the LPI2C target matched the last address byte. When SCFGR1[RSCFG] = 1, this flag is set on any repeated Start condition. This flag is not set when the target first detects a Start condition. **NOTE** This field behaves differently for register reads and writes. When reading 0b - No repeated Start detected 1b - Repeated Start detected When writing 0b - No effect 1b - Clear the flag
7-4 —	Reserved
3 TAF	Transmit ACK Flag Indicates whether a transmit ACK or NACK is required. You can clear this flag by writing to Target Transmit ACK (STAR). 0b - Not required

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	1b - Required
2 AVF	Address Valid Flag Indicates whether the contents of Target Address Status (SASR) are valid. You can clear this flag by reading SASR. When SCFGR1[RXCFG] = 1, this flag is also cleared by reading Target Receive Data (SRDR). 0b - Not valid 1b - Valid
1 RDF	Receive Data Flag Indicates whether receive data is ready. You can clear this flag by reading Target Receive Data (SRDR). When SCFGR1[RXCFG] = 1, this flag is not cleared when reading Target Receive Data (SRDR) if SSR[AVF] = 1. 0b - Not ready 1b - Ready
0 TDF	Transmit Data Flag Indicates whether transmit data has been requested. This flag is cleared by writing to Target Transmit Data (STDR). When SCFGR1[TXCFG] = 0, if a NACK, repeated Start, or Stop condition is detected, this flag is also cleared. 0b - Transmit data not requested 1b - Transmit data is requested

31.7.1.22 Target Interrupt Enable (SIER)

Offset

Register	Offset
SIER	118h

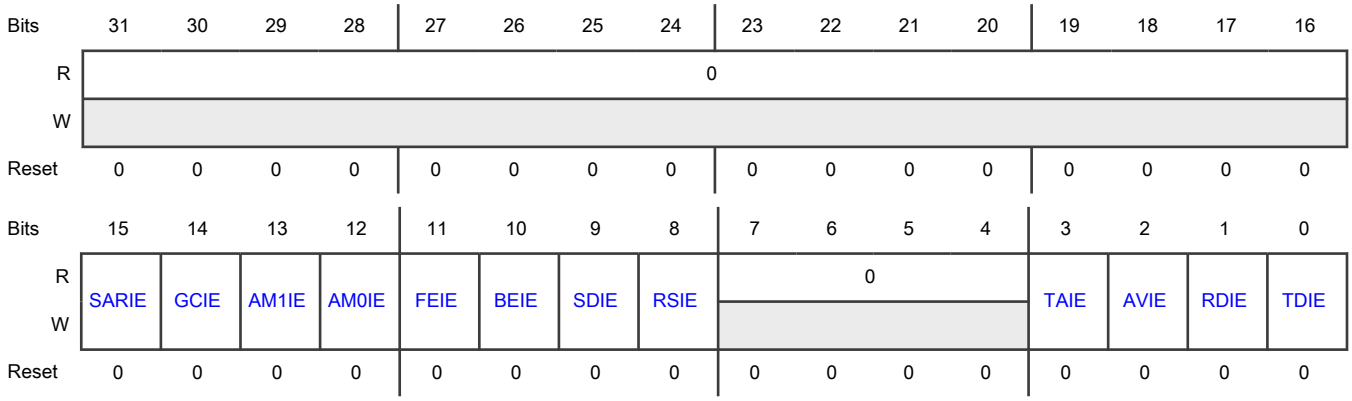
Function

Contains transmit and receive data interrupt enables, start and stop detect interrupt enables, and other target interrupt enables.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram



Fields

Field	Function
31-16 —	Reserved
15 SARIE	SMBus Alert Response Interrupt Enable Enables interrupt for SMBus alert response. 0b - Disable 1b - Enable
14 GCIE	General Call Interrupt Enable Enables interrupt for general call. 0b - Disabled 1b - Enabled
13 AM1IE	Address Match 1 Interrupt Enable Enables interrupt for address match 1. 0b - Disable 1b - Enable
12 AM0IE	Address Match 0 Interrupt Enable Enables interrupt for address match 0. 0b - Disable 1b - Enable
11 FEIE	FIFO Error Interrupt Enable Enables interrupt for FIFO error. 0b - Disable 1b - Enable

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
10 BEIE	Bit Error Interrupt Enable Enables interrupt for bit error. 0b - Disable 1b - Enable
9 SDIE	Stop Detect Interrupt Enable Enables interrupt for Stop detection. 0b - Disable 1b - Enable
8 RSIE	Repeated Start Interrupt Enable Enables interrupt for repeated start. 0b - Disable 1b - Enable
7-4 —	Reserved
3 TAIE	Transmit ACK Interrupt Enable Enables interrupt for transmit ACK. 0b - Disable 1b - Enable
2 AVIE	Address Valid Interrupt Enable Enables interrupt for valid address. 0b - Disable 1b - Enable
1 RDIE	Receive Data Interrupt Enable Enables interrupt for receive data. 0b - Disable 1b - Enable
0 TDIE	Transmit Data Interrupt Enable Enables interrupt for transmit data. 0b - Disable 1b - Enable

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

31.7.1.23 Target DMA Enable (SDER)

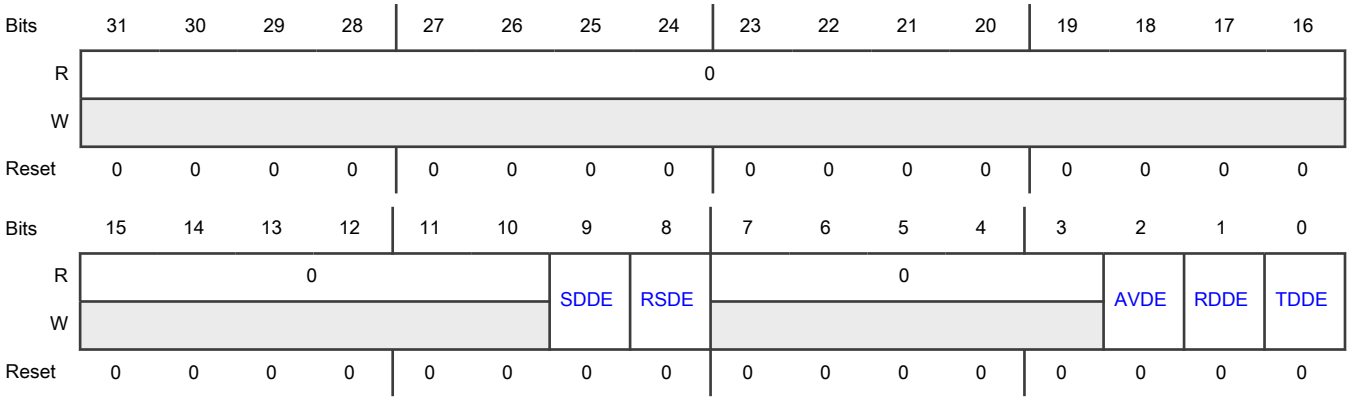
Offset

Register	Offset
SDER	11Ch

Function

Contains the transmit, request, and receive enables for DMA.

Diagram



Fields

Field	Function
31-10 —	Reserved
9 SDDE	Stop Detect DMA Enable Enables DMA end-of-packet processing on stop detection. Reading Target Receive Data (SRDR) when it is empty: • Generates a DMA end-of-packet response. • Returns 0000_40FFh. • Writes 0 to SSR[SDF]. 0b - Disable 1b - Enable
8 RSDE	Repeated Start DMA Enable Enables DMA end-of-packet processing on repeated start. Reading Target Receive Data (SRDR) when it is empty: • Generates a DMA end-of-packet response.

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	• Returns 0000_40FFh. • Writes 0 to SSR[RDF]. 0b - Disable 1b - Enable
7-3 —	Reserved
2 AVDE	Address Valid DMA Enable Enables address valid DMA request. The address valid DMA request is shared with the receive data DMA request. If both are enabled, write 1 to SCFGR1[RXCFG] to allow the DMA to read the address from Target Receive Data (SRDR). 0b - Disable 1b - Enable
1 RDDE	Receive Data DMA Enable Enables receive data for DMA. 0b - Disable DMA request 1b - Enable DMA request
0 TDDE	Transmit Data DMA Enable Enables transmit data for DMA. 0b - Disable 1b - Enable

31.7.1.24 Target Configuration 0 (SCFGR0)

Offset

Register	Offset
SCFGR0	120h

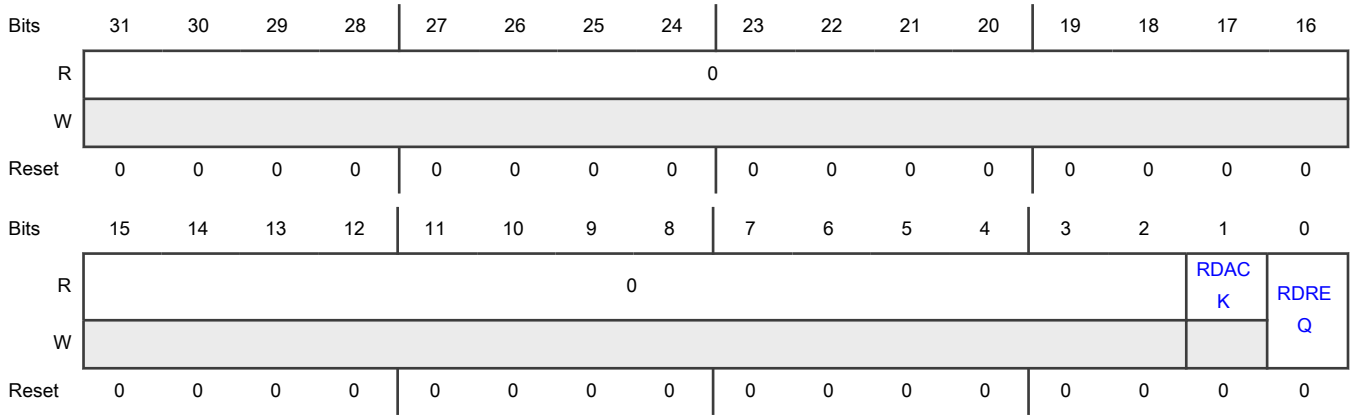
Function

Configures the read request feature.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram



Fields

Field	Function
31-2 —	Reserved
1 RDACK	Read Acknowledge Flag Indicates whether a Start then Stop sequence with one SCL pulse between them acknowledged the read request while SCFGR0[RDREQ] = 1. You can clear this flag by writing 0 to SCFGR0[RDREQ]. 0b - Read Request not acknowledged 1b - Read Request acknowledged
0 RDREQ	Read Request Enables read request. When the I2C bus is idle, writing 1 to this field causes the LPI2C target to drive SDA low, triggering a Start condition. The LPI2C target releases SDA on the next falling edge of SCL or when you write 0 to this field. To initiate a second read request (for example, following I2C bus activity that did not acknowledge the request), write 0 then 1 to this field. When the I2C bus is busy, writing to this register always writes 0 to this field. 0b - Disable 1b - Enable

31.7.1.25 Target Configuration 1 (SCFGR1)

Offset

Register	Offset
SCFGR1	124h

All information provided in this document is subject to legal disclaimers.

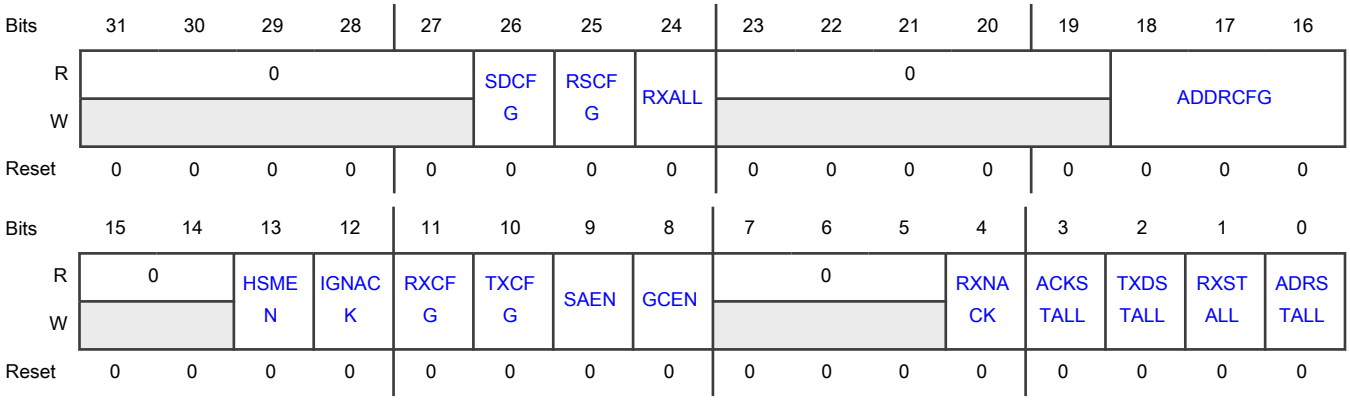
© 2026 NXP B.V. All rights reserved.

Function

Configures various aspects of the target.

Write to this register only when the I2C target is disabled.

Diagram



Fields

Field	Function
31-27 —	Reserved
26 SDCFG	Stop Detect Configuration Configures the conditions that set SSR[SDF]. 0b - Any Stop condition following an address match 1b - Any Stop condition
25 RSCFG	Repeated Start Configuration Configures the conditions that set SSR[RSF]. 0b - Any repeated Start condition following an address match 1b - Any repeated Start condition
24 RXALL	Receive All Enables receive-all functionality. When enabled, the LPI2C target stores all addresses on the I2C bus in Target Address Status (SASR) and all data on the I2C bus in Target Receive Data (SRDR). However, the LPI2C target does not drive SDA (for ACK or target-transmit transfer) unless there is an address match. The LPI2C target can drive SCL if clock stretching is enabled. When this field is 1, the LPI2C target only supports 7-bit addressing modes (you must configure SCFGR1[ADDRCFG] for 7-bit address match). Software can support 10-bit addresses, however. The first byte of a 10-bit address is saved to Target Address Match (SAMR). The second byte is saved as the first data byte in Target Receive Data (SRDR).

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	Use this field to aid debugging of the I2C bus by saving all data on the bus without altering the state of the bus. When this field is 1, it is recommended to also write 1 to SCFGR1[RSCFG] and SCFGR1[SDCFG] so you can track the full state of the I2C bus. 0b - Disable 1b - Enable
23-19 —	Reserved
18-16 ADDRCFG	Address Configuration Configures the condition that causes an address to match. 000b - Address match 0 (7-bit) 001b - Address match 0 (10-bit) 010b - Address match 0 (7-bit) or address match 1 (7-bit) 011b - Address match 0 (10-bit) or address match 1 (10-bit) 100b - Address match 0 (7-bit) or address match 1 (10-bit) 101b - Address match 0 (10-bit) or address match 1 (7-bit) 110b - From address match 0 (7-bit) to address match 1 (7-bit) 111b - From address match 0 (10-bit) to address match 1 (10-bit)
15-14 —	Reserved
13 HSMEN	HS Mode Enable Enables detection of the HS mode controller code of target address 0000_1XX, but does not cause an address match on this code. When this field is 1 and any HS mode controller code is detected, SCR[FILTEN] and SCFGR1[ACKSTALL] are ignored until the next Stop condition is detected. 0b - Disable 1b - Enable
12 IGNACK	Ignore NACK Determines whether the target ends transfer when a NACK condition is detected. When this field is 1, the LPI2C target continues transfers after a NACK is detected. This field is required to be 1 in Ultra-Fast mode. 0b - End transfer on NACK 1b - Do not end transfer on NACK
11 RXCFG	Receive Data Configuration Configures which data is returned and which flags are cleared when reading Target Receive Data (SRDR).

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	When this field is 0, reading SRDR returns received data and clears SSR[RDF]. When this field is 1, reading SRDR: • Returns the value of Target Address Status (SASR) and clears SSR[AVF] when SSR[AVF] is set. • Returns received data and clears SSR[RDF] when SSR[AVF] is not set. 0b - Return received data, clear SSR[RDF] 1b - Return SASR and clear SSR[AVF] when SSR[AVF] is set, return received data and clear SSR[RDF] when SSR[AFV] is not set
10 TXCFG	Transmit Flag Configuration Determines which conditions set SSR[TDF]. This field always becomes 1 before a NACK is detected at the end of a target-transmit transfer. This change can cause an extra word to be written to the transmit data FIFO. When this field is 0, Target Transmit Data (STDR) is automatically emptied when a target-transmit transfer is detected. SSR[TDF] is set when a target-transmit transfer is detected, and SSR[TDF] is cleared at the end of the target-transmit transfer. When this field is 1, SSR[TDF] is set when STDR is empty, and SSR[TDF] is cleared when STDR is full. This setting allows STDR to be filled before a target-transmit transfer is detected. However, it can cause STDR to be written before a NACK is detected on the last byte of a target-transmit transfer. 0b - SSR[TDF] is set only during a target-transmit transfer when STDR is empty 1b - SSR[TDF] is set whenever STDR is empty
9 SAEN	SMBus Alert Enable Enables a match on an SMBus alert. 0b - Disable 1b - Enable
8 GCEN	General Call Enable Enables a general call address. 0b - Disable 1b - Enable
7-5 —	Reserved
4 RXNACK	Receive NACK Determines whether to override the setting of STAR[TXNACK] when the LPI2C receives a matching address during an overrun. When this field is 1, the LPI2C target responds with a NACK under the following conditions:

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	• SSR[AVF] would be set due to matching an address, but that flag is already 1 (address overrun). • SSR[RDF] would be set due to receiving data, but that flag is already 1 (receive data overrun). 0b - ACK or NACK always determined by STAR[TXNACK] 1b - NACK always generated on address overrun or receive data overrun, otherwise ACK or NACK is determined by STAR[TXNACK]
3 ACKSTALL	ACK SCL Stall Enables SCL clock stretching during target-transmit address bytes and target-receiver address and data bytes, so you can write to Target Transmit ACK (STAR) before the ACK or NACK is transmitted. Clock stretching occurs when transmitting the ninth bit, and is therefore not compatible with HS mode. If this field is 1: • You do not need to write 1 to SCFGR1[RXSTALL] or SCFGR1[ADRSTALL]. • When there is an address match on the first byte of a 10-bit address, SSR[AVF] is set, allowing you to read the received address before writing to Target Transmit ACK (STAR). 0b - Disable 1b - Enable
2 TXDSTALL	Transmit Data SCL Stall Enables SCL clock stretching when SSR[TDF] = 1 during a target-transmit transfer. Clock stretching occurs following the ninth bit, and is therefore compatible with HS mode. 0b - Disable 1b - Enable
1 RXSTALL	RX SCL Stall Enables SCL clock stretching when SSR[RDF] = 1 during a target-receive transfer. Clock stretching occurs following the ninth bit, and is therefore compatible with HS mode. 0b - Disable 1b - Enable
0 ADRSTALL	Address SCL Stall Enables SCL clock stretching when SSR[AVF] = 1. Clock stretching only occurs following the ninth bit, and is therefore compatible with HS mode. 0b - Disable 1b - Enable

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

31.7.1.26 Target Configuration 2 (SCFGR2)

Offset

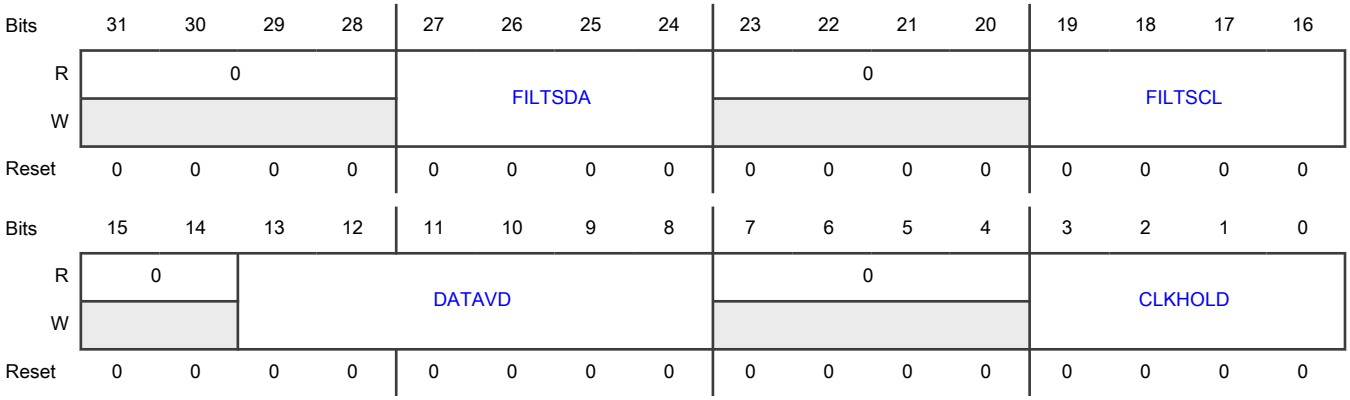
Register	Offset
SCFGR2	128h

Function

Configures data valid delay, clock hold time, and glitch filters for SDA and SCL.

Write to this register only when the I2C target is disabled.

Diagram



Fields

Field	Function
31-28 —	Reserved
27-24 FILTSDA	Glitch Filter SDA Configures the I2C target digital glitch filters for SDA input. Writing 0 to this field disables the glitch filter. Glitches equal to or less than the number of cycles defined by this field are filtered out and ignored. The latency through the glitch filter is equal to the number of cycles defined by this field + 3. The latency must be configured to be less than the minimum SCL low or high period. MCFGR1[PRESCALE] does not affect the glitch filter cycle count, and the glitch filter cycle count is disabled in HS mode.
23-20 —	Reserved
19-16	Glitch Filter SCL

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
FILTSCL	Configures the I2C target digital glitch filters for SCL input. Writing 0 to this field disables the glitch filter. Glitches equal to or less than the number of cycles defined by this field are filtered out and ignored. The latency through the glitch filter is equal to the number of cycles defined by this field + 3. The latency must be configured to be less than the minimum SCL low or high period. MCFGR1[PRESCALE] does not affect the glitch filter cycle count, and the glitch filter cycle count is disabled in HS mode.
15-14 —	Reserved
13-8 DATAVD	Data Valid Delay Configures the SDA data valid delay time for the I2C target, which is equal to FILTSCL + DATAVD + 3 cycles. The data valid delay must be configured to be less than the minimum SCL low period. MCFGR1[PRESCALE] does not affect the I2C target data valid delay time, and the I2C target data valid delay time is disabled in HS mode.
7-4 —	Reserved
3-0 CLKHOLD	Clock Hold Time Configures the minimum clock hold time for the I2C target, when clock stretching is enabled. The minimum hold time is equal to the number of cycles defined by this field + 3. The FILTSDA/FILTSCL state that writing 0 will disable the filter and CLKHOLD is similar (writing 0 will disable the clock stretching extension). MCFGR1[PRESCALE] does not affect the I2C target clock hold time, and the I2C target clock hold time is disabled in HS mode.

31.7.1.27 Target Address Match (SAMR)

Offset

Register	Offset
SAMR	140h

Function

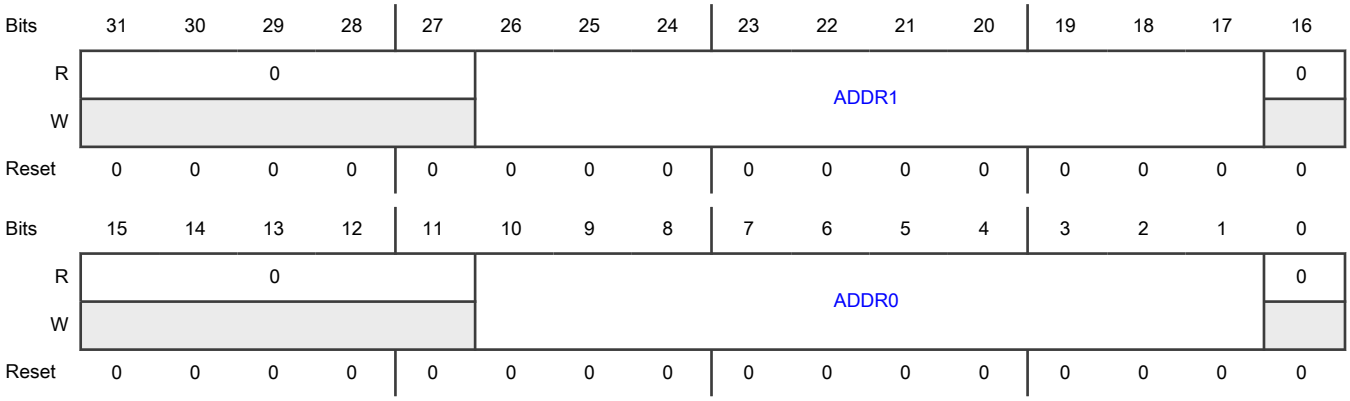
Contains address values for received target match comparison.

Write to this register only when the I2C target is disabled.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram



Fields

Field	Function
31-27 —	Reserved
26-17 ADDR1	Address 1 Value Contains the value of address 1, which is compared to the received address to detect the target address. In 10-bit mode, the first address byte is compared to {11110, ADDR1[26:25]} and the second address byte is compared to ADDR1[24:17]. In 7-bit mode, the address is compared to ADDR1[23:17].
16-11 —	Reserved
10-1 ADDR0	Address 0 Value Contains the value of address 0, which is compared to the received address to detect the target address. In 10-bit mode, the first address byte is compared to {11110, ADDR0[10:9]} and the second address byte is compared to ADDR0[8:1]. In 7-bit mode, the address is compared to ADDR0[7:1].
0 —	Reserved

31.7.1.28 Target Address Status (SASR)

Offset

Register	Offset
SASR	150h

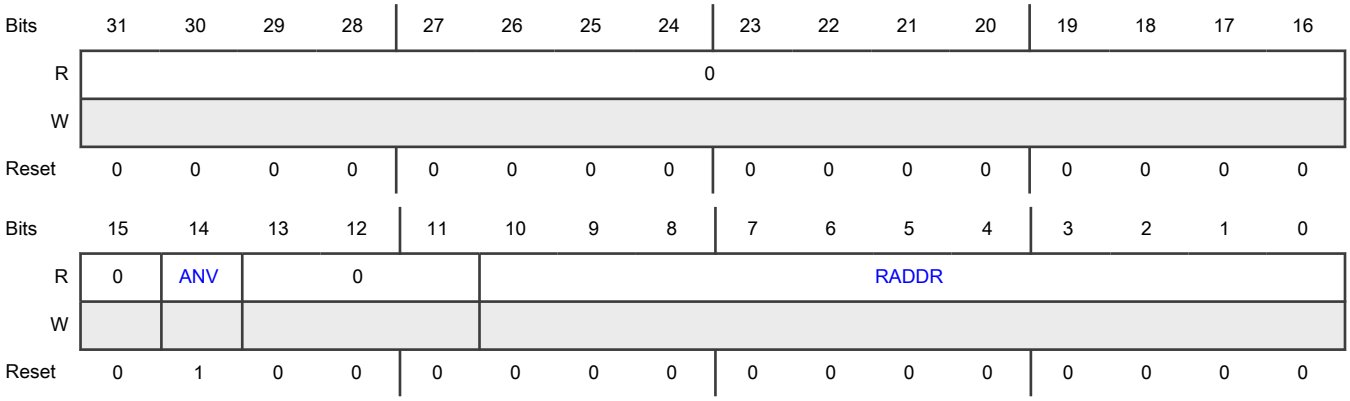
All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Function

Contains the received address and its validity.

Diagram



Fields

Field	Function
31-15 —	Reserved
14 ANV	Address Not Valid Indicates whether SASR[RADDR] is valid. 0b - Valid 1b - Not valid
13-11 —	Reserved
10-0 RADDR	Received Address Contains the received address. Updates whenever SSR[AM0F] or SSR[AM1F] is set. Reading Target Address Status (SASR) clears SSR[AM0F] and SSR[AM1F]. In 7-bit mode, the address byte is stored in RADDR[7:1]. The Read-or-Write bit is stored in RADDR[0]. In 10-bit mode, the first address byte is {11110, RADDR[10:9], RADDR[0]} and the second address byte is RADDR[8:1]. The Read-or-Write bit is therefore always stored in RADDR[0]. When SCFGR1[ACKSTALL] = 1, if the first address byte matches in 10-bit mode, the first address byte is stored in RADDR[7:0] so you can read this field before writing the Transmit ACK. If the second address byte matches, this field is then updated with the full 10-bit address.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

31.7.1.29 Target Transmit ACK (STAR)

Offset

Register	Offset
STAR	154h

Function

Configures choice of ACK or NACK on each received word.

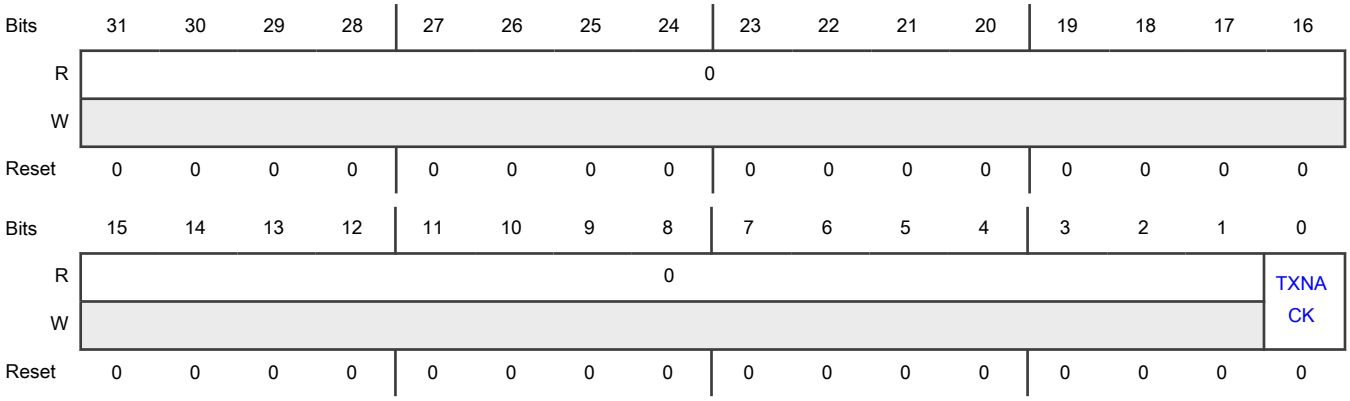
You can write to this register only when SCFGR1[ACKSTALL] = 1.

SCFGR1[ACKSTALL] enables clock stretching during the ACK-or-NACK bit slot. During this time, you can write to this register.

The logic ensures that the clock stretching continues for at least one bus clock cycle after this register is updated.

This clock stretching time can be extended via SCFGR2[CLKHOLD].

Diagram



Fields

Field	Function
31-1 —	Reserved
0 TXNACK	Transmit NACK Selects whether transmit ACK (logic 0) or NACK (logic 1) is returned on the bus by the I2C target after receiving each word. • When SCFGR1[ACKSTALL] = 1, a transmit NACK signal must be written once for each matching address byte and each received word. SCFGR1[ACKSTALL] must be 1, because that setting stalls the data transfer until software reads the received word (and determines whether to respond with an ACK or NACK). • To configure the default (ACK or NACK), you can write to this field when LPI2C target is disabled or idle. 0b - Transmit ACK 1b - Transmit NACK

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

31.7.1.30 Target Transmit Data (STDR)

Offset

Register	Offset
STDR	160h

Function

Contains the I2C target data to transmit.

Clock stretching (enabled or disabled) affects when the transmit data is transferred. SCFGR1[TXDSTALL] enables clock stretching during the first data bit of a target-transmit transfer.

If clock stretching is enabled (SCFGR1[TXDSTALL] = 1), the transmit data transfer is stalled until this register is updated. Clock stretching is extended by at least 1 bus clock cycle after this register is updated. Clock stretching can be delayed further by using SCFGR2[CLKHOLD].

If clock stretching is disabled (SCFGR1[TXDSTALL] = 0), the transmit data must be written before the start of the target-transmit transfer, otherwise SSR[FEF] is set.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R																
W	0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R																
W	0								DATA							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-8 —	Reserved
7-0 DATA	Transmit Data Contains the I2C target data to transmit. Writing data to this register stores I2C target transmit data in this register.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

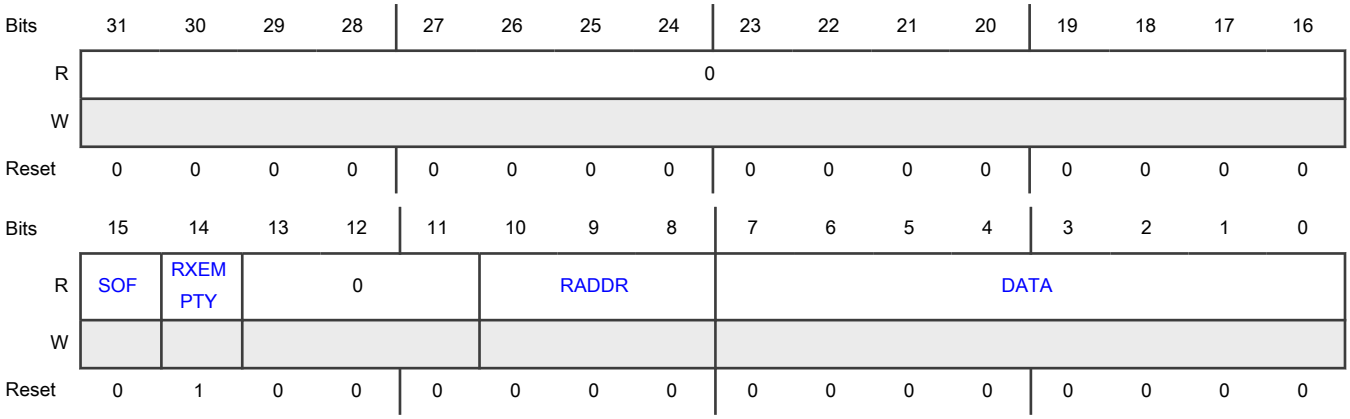
31.7.1.31 Target Receive Data (SRDR)

Offset

Register	Offset
SRDR	170h

Function
Contains status of target receive data transfer.

Diagram



Fields

Field	Function
31-16 —	Reserved
15 SOF	Start of Frame Indicates whether this data word is the first data word since a (repeated) Start or Stop condition. 0b - Not first 1b - First
14 RXEMPTY	Receive Empty Indicates whether this register is empty. 0b - Not empty 1b - Empty
13-11 —	Reserved
10-8	Received Address

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
RADDR	Contains the address received by the IC2 target. When both SCFGR1[RXCFG] and SSR[AVF] are 1, bits [10:8] of SASR[RADDR] are returned. Otherwise, this field returns zero.
7-0 DATA	Received Data Contains the data received by the I2C target. When both SCFGR1[RXCFG] and SSR[AVF] are 1, bits [7:0] of SASR[RADDR] are returned.

31.7.1.32 Target Receive Data Read Only (SRDROR)

Offset

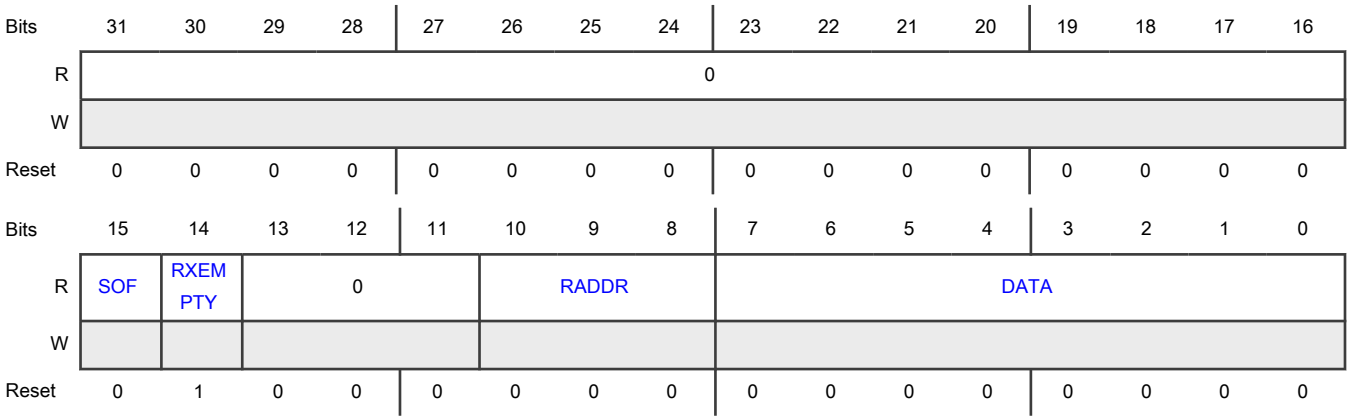
Register	Offset
SRDROR	178h

Function

Contains the data received by the I2C target.

Reading this register returns the data received by the I2C target, but does not pull the data from the register.

Diagram



Fields

Field	Function
31-16 —	Reserved
15 SOF	Start of Frame Indicates whether this data word is the first data word since a (repeated) Start or Stop condition.

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	0b - Not the first 1b - First
14 RXEMPTY	Receive Empty Indicates whether Target Receive Data (SRDR) is empty. 0b - Not empty 1b - Empty
13-11 —	Reserved
10-8 RADDR	Received Address Contains address received by the LPI2C target. When both SCFGR1[RXCFG] and SSR[AVF] are 1, bits [10:8] of SASR[RADDR] are returned. Otherwise, this field returns zero.
7-0 DATA	Receive Data Contains data received by the LPI2C target. When both SCFGR1[RXCFG] and SSR[AVF] are 1, bits [7:0] of SASR[RADDR] are returned.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Chapter 32
Low Power Universal Asynchronous Receiver/ Transmitter (LPUART)

32.1 Chip-specific LPUART information

Table 185. Reference links to related information

Topic	Related module	Reference
Full description	LPUART	LPUART
System memory map		Memory map
Clocking		Clock distribution
Power Management		Power Management
Signal multiplexing	Port control	Signal multiplexing

32.1.1 Module instances

This chip contains four instances of LPUART: LPUART0, LPUART1, LPUART2 and LPUART3.

32.2 Overview

LPUART provides asynchronous, serial communication capabilities with external devices. It supports the non-return-to-zero (NRZ) encoding format and infrared data association (IrDA)-compatible, low-speed serial infrared (SIR) protocol. LPUART can continue operating when the processor is in Low-Power mode, if an appropriate peripheral clock is available.

32.2.1 Block diagram

Figure 134 shows the transmitter portion of LPUART.

All information provided in this document is subject to legal disclaimers.
© 2026 NXP B.V. All rights reserved.

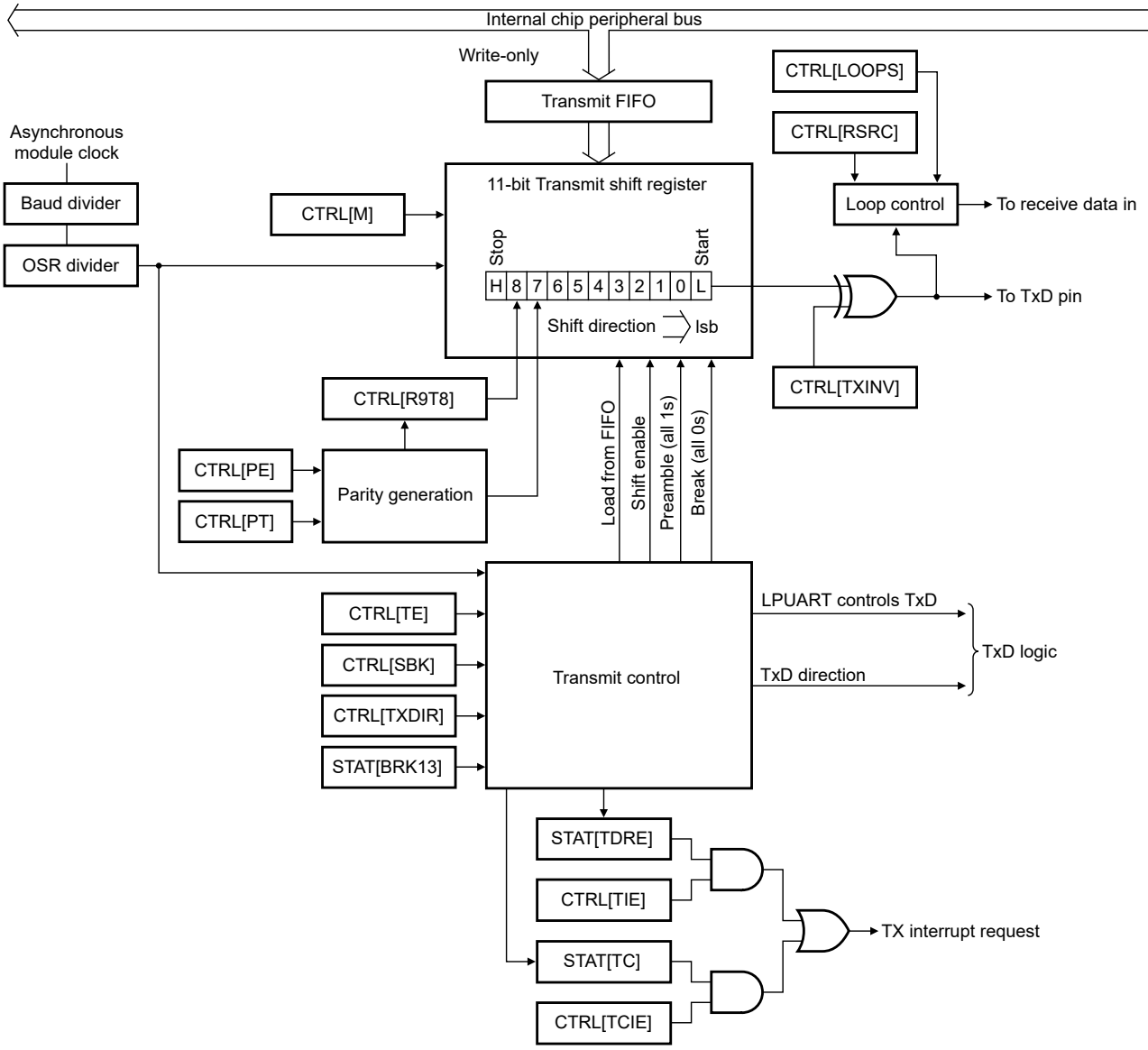


Figure 134. Transmitter block diagram

Figure 135 shows the receiver portion of LPUART.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

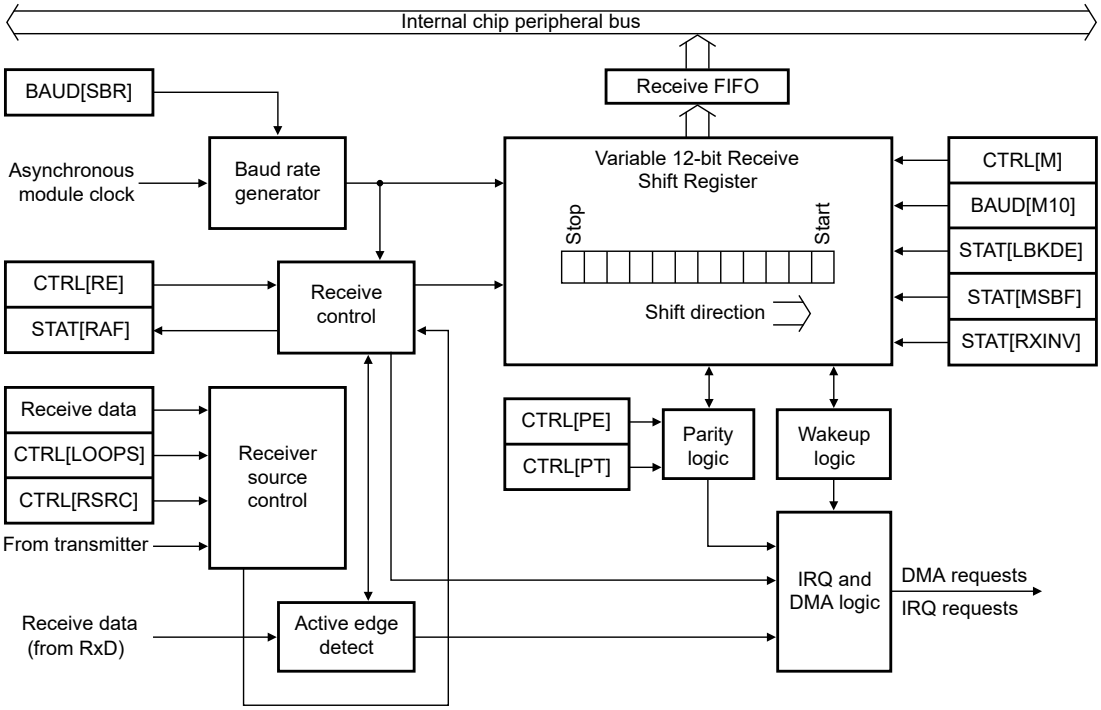


Figure 135. Receiver block diagram

32.2.2 Features

- Full-duplex, standard NRZ format
- Programmable baud rates (13-bit modulo divider) with a configurable oversampling ratio (OSR) from 4× to 32×
- Support of 4800 and 9600 baud rate with 32.768 KHz operating clock (asynchronous module clock from external sources)
- Asynchronous operation of transmit and receive baud rates with respect to the bus clock:
  - Baud rate can be configured independently of the bus clock frequency.
  - Operation in Low-Power modes is supported.
- Programmable option for early and mid point sampling of incoming asynchronous data
- Interrupt, DMA, or polled operations:
  - Transmit data empty and transmission complete
  - Receive data full
  - Receive overrun, parity error, framing error, and noise error
  - Idle receiver detect
  - Active edge on receive pin
  - Break detect supporting LIN
  - Receive data match
- Hardware parity generation and checking
- Programmable 7-bit, 8-bit, 9-bit, or 10-bit character length
- Programmable 1-bit or 2-bit stop bits

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

- Support for three receiver wake-up methods:
  - Idle line wake-up
  - Address mark wake-up
  - Receive data match
- Automatic address matching to reduce ISR overhead:
  - Address mark matching
  - Idle line address matching
  - Address match start, address match end
- Optional 13-bit and 11-bit break character generation
- Configurable idle length detection supporting 1, 2, 4, 8, 16, 32, 64, or 128 idle characters
- Selectable transmitter output and receiver input polarity
- Hardware flow control support for request to send (RTS) and clear to send (CTS) signals
- Selectable IrDA 1.4 return-to-zero-inverted (RZI) format with a programmable pulse width
- Independent FIFO structure for transmit and receive functions:
  - Separate configurable watermarks for receive and transmit requests
  - Option for receiver to assert request after a configurable number of idle characters, if receive FIFO is not empty
- Support swapping of TXD and RXD

32.3 Functional description

LPUART supports full-duplex, asynchronous, NRZ serial communication and comprises a baud rate generator, transmitter, and receiver block. The transmitter and receiver operate independently, although they use the same baud rate generator. The following sections describe all LPUART blocks.

32.3.1 Baud rate generation

A 13-bit modulus counter in the baud rate generator derives the baud rate for both the receiver and transmitter. The value, ranging from 1 to 8191, written to BAUD[SBR] determines the baud clock divisor for the asynchronous LPUART baud clock. The baud rate clock drives the receiver, while a bit clock, generated from the baud rate clock divided by the OSR, drives the transmitter. Depending on the OSR, the receiver has an acquisition rate of 4 to 32 samples per bit time. LPUART requires BAUD[SBR] and BAUD[OSR] to accurately match the baud rate requirement. There is a relationship between the required baud rate and the input clock frequency, SBR, and OSR. See the below figure for details.

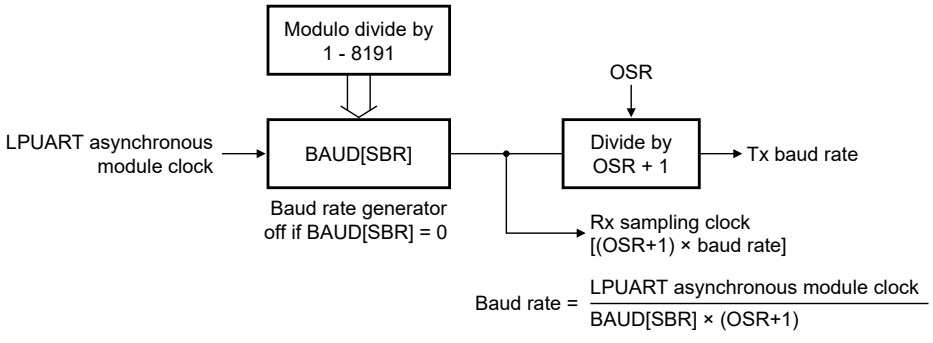


Figure 136. Baud rate generation

Baud rate generation is subject to these sources of error:

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

- Integer division of the asynchronous LPUART baud clock may not give the exact target frequency.
- Synchronization with the asynchronous LPUART baud clock can lead to a phase shift.

Baud rate generation is a free-running counter that continues whenever the transmitter or receiver is enabled. The transmitter bit clock continues whenever the transmitter is enabled; each transmitted character aligns to the next edge of the transmit bit clock.

In general, configuring OSR for a higher ratio and/or sampling on both edges of the clock slightly improves LPUART's tolerance to baud rate mismatch between the received data and LPUART configured baud rate. However, the three data samples in each bit (see Data sampling technique) are also closer together, which may impact noise sensitivity.

32.3.2 Baud rate tolerance

A transmitting device may operate at a baud rate below or above that of the receiver.

Accumulated bit time misalignment can cause one of the three stop bit data samples to fall outside the actual stop bit. A noise error will occur if the three samples are not all the same logical values. A framing error will occur if the receiver clock is misaligned in such a way that the majority of the three stop bit samples are a logic zero.

As the receiver samples an incoming frame, it may resynchronize the oversampling clock on any valid falling edge within the frame. Resynchronization within frames will correct a misalignment between transmitter bit times and receiver bit times.

In general, increasing the number of samples per bit will increase the baud rate tolerance and decreasing the number of samples per bit will reduce the baud rate tolerance. Note that since LPUART implements triple voting on consecutive receive data samples, increasing the number of samples per bit will move those samples closer together which would reduce the width of noise that can be filtered by the triple voting logic.

32.3.3 Calculating baud rate tolerance

Using the following definitions:

- SAM is the number of sample points per bit (valid range from 8 to 32; equal to (OSR + 1) × (BOTHEDGE + 1)).
- BIT is the number of bits in a character including start, data and stop bits (valid range from 9 to 13).

The ideal baud rate tolerance can be calculated as follows:

- Slow data rate tolerance = ((SAM ÷ 2) - 1) ÷ ((SAM × BIT) - (SAM ÷ 2) + 2)
- Fast data rate tolerance = ((SAM ÷ 2) - 2) ÷ (SAM × BIT)

As an example, if configured for 8-bit data, 1 stop bit (BIT = 10) and with OSR=0x7 and BOTHEDGE = 1 (SAM = 16):

- Slow data rate tolerance = (8 - 1) ÷ (160 - 8 + 2) = 7 ÷ 154 = 4.54%
- Fast data rate tolerance = (8 - 2) ÷ 160 = 6 ÷ 160 = 3.75%

If configured for 9-bit data with 1 stop bit (BIT = 11) with same oversampling configuration, then:

- Slow data rate tolerance = (8 - 1) ÷ (176 - 8 + 2) = 7 ÷ 170 = 4.12%
- Fast data rate tolerance = (8 - 2) ÷ 176 = 6 ÷ 176 = 3.41%

NOTE

Additional factors can contribute to a lower baud rate tolerance than the ideal. These include clock uncertainty or jitter on the LPUART functional clock source, differences in rise and fall times on the transmitter output and synchronization of the external receive pin to the local LPUART functional clock.

32.3.4 Support of 32.768 KHz asynchronous module clock

If the input clock frequency is very low, for example, the input clock is supplied by an external off-chip RTC oscillator in the range of KHz, the baud rate calculation formula does not provide accurate baud rate in case of both transmitting and receiving UART packets. As a result, chances of character slip are there within a single UART packet.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

To tackle this LPUART uses a special configuration field to enable the capability of modifying the programmed oversampling ratio by 0.5. This allows LPUART to closely match the baud rate values even when a very slow clock is used as input. LPUART supports limited baud rates as shown in the table below when the asynchronous module clock (sampling clock) is 32.768 KHz. The following table provides the configurations for supported baud rates. When BAUD[MSEL32K] is 0, the LPUART baud rate generation works normally as mentioned in Baud rate generation.

Table 186. Configuration for supported baud rates

External UART clock frequency (KHz)	Supported baud rate (Kbps)	BAUD[MSEL32K]	BAUD[BOTHEDGE]	BAUD[SBR]	BAUD[OSR]
32.768	9.6	1	1	1	4
32.768	4.8	1	1	1	7

32.3.5 Transmitter functional description

This section describes the functioning of the LPUART transmitter, as shown in the transmitter portion of Block diagram, as well as specialized functions for sending break and idle characters.

The transmitter output (TXD) idle state defaults to logic high; the transmitter output is inverted when you write 1 to CTRL[TXINV], which becomes 0 following reset. You can enable the transmitter by writing 1 to CTRL[TE]. This queues a preamble character that is one full character frame of the Idle state. The transmitter then remains idle until data is available in the transmit FIFO and programs store data in the transmit FIFO by writing to Data (DATA).

The central element of the LPUART transmitter is the transmit shift register that is 9-bit to 13-bit long depending on the settings of CTRL[M], CTRL[M7], BAUD[M10], and BAUD[SBNS]. Going forward in this discussion, assume that CTRL[M], CTRL[M7], BAUD[M10], and BAUD[SBNS] are 0, selecting the normal 8-bit Data mode, in which the shift register holds a start bit, eight data bits, and a stop bit. When the transmit shift register is available for a new character, the value waiting in transmit FIFO is transferred to the transmit shift register, synchronized with the baud rate clock, and STAT[TDRE] becomes 1 to indicate that another character may be written to the transmit FIFO at Data (DATA).

If no new character is waiting in the transmit FIFO after a stop bit is shifted out of the TXD pin, the transmitter sets the transmit complete flag and enters an idle mode, with TXD high, waiting for more characters to transmit.

Writing 0 to CTRL[TE] does not immediately disable the transmitter. The current transmit activity in progress must first be completed (that could include a data character, idle character, or break character), although the transmitter does not start transmitting another character.

32.3.5.1 Break character length

CTRL[SBK] sends break characters, originally used to gain the attention of old teletype receivers. Break characters are a full character time of logic 0, 9-bit to 12-bit times, including the start and stop bits. You can enable a longer break of 13-bit times by writing 1 to STAT[BRK13]. Normally, a program waits for STAT[TDRE] to become 1 to indicate that the last character of a message has moved to the transmit shifter. Next, the program writes 1 and then writes 0 to CTRL[SBK]. This action queues a break character to be sent as soon as the shifter is available. If CTRL[SBK] remains 1 when the queued break moves into the shifter, synchronized with the baud rate clock, an additional break character is queued. When LPUART is the receiving module, it receives a break character as 0s in all data bits and a framing error (STAT[FE] = 1) is detected.

You can also transmit a break character by writing to Data (DATA) with DATA[FRETSC] = 1 and the data bits clear. This supports transmitting the break character as part of the normal data stream and also allows DMA to transmit a break character.

When idle line wake-up is used, a full character time of idle (logic 1) is needed between messages to wake up any sleeping receivers. Normally, a program waits for STAT[TDRE] to become 1 to indicate that the last character of a message has moved to the transmit shifter. Next, write 0 and then write 1 to CTRL[TE]. This action queues an idle character to be sent as soon as the shifter is available. As long as the character in the shifter does not finish while CTRL[TE] becomes 0, the LPUART transmitter does not release control of the TXD pin.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

You can also write to Data (DATA) to transmit an idle character, with DATA[FRETSC] and DATA[R9T9] = 1 and the values of all the other fields = 0. This supports transmitting the idle character as part of the normal data stream and also allows DMA to transmit an idle character.

As shown in the following table, STAT[BRK13], CTRL[M], CTRL[M7], BAUD[M10], and BAUD[SBNS] affect the length of the break character.

Table 187. Break character length

STAT[BRK13]	CTRL[M]	BAUD[M10]	CTRL[M7]	BAUD[SBNS]	Break character length (in bit times)
0	0	0	0	0	10
0	0	0	0	1	11
0	0	0	1	0	9
0	0	0	1	1	10
0	1	0	—	0	11
0	1	0	—	1	12
0	—	1	—	0	12
0	—	1	—	1	13
1	0	0	0	0	13
1	0	0	0	1	13
1	0	0	1	0	12
1	0	0	1	1	12
1	1	0	—	0	14
1	1	0	—	1	14
1	—	1	—	0	15
1	—	1	—	1	15

32.3.5.2 Hardware flow control

The transmitter supports hardware flow control by gating the transmission with the value of CTS_B. If the CTS operation is enabled, the character is transmitted when CTS_B is asserted. If CTS_B is deasserted in the middle of a transmission with characters remaining in the transmitter FIFO, the character in the transmit shift register is complete. Any characters in the FIFO wait for CTS_B to assert again, and TXD remains in the mark state (idle state) until CTS_B is reasserted. The CTS_B pin must assert for longer than one bit period to guarantee that a new transmission is started when the transmitter is idle with CTS.

If the CTS operation is disabled, the transmitter ignores the state of CTS_B.

The transmitter's CTS_B signal can be enabled even if the same LPUART receiver's RTS_B signal is disabled.

In case of sending break characters, if the CTS operation is enabled, you can send break characters (CTRL[SBK] = 1), irrespective of the state of the CTS_B signal.

32.3.5.3 Transceiver driver enable

The transmitter can use RTS_B as an enable signal for the driver of an external transceiver. See Transceiver driver enable using RTS_B for details. If the RTS operation is enabled, when a character is placed into an empty transmit shift register, RTS_B asserts 1-bit time before the start bit is transmitted. RTS_B remains asserted for the whole time that the transmit shift register has any characters. RTS_B deasserts 1-bit time after all characters in the transmit FIFO and shift register are completely sent, including the

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

last stop bit. In other words, when RTS_B is used as a transceiver enable, RTS_B asserts 1-bit time before the transmitter starts transmitting and negates 1-bit time after the transmitter goes idle.

Transmitting a break character also asserts RTS_B, with the same assertion and deassertion timing as having a character in the transmit shift register.

The transmitter's RTS_B signal asserts only when the transmitter is enabled. However, the transmitter's RTS_B signal is unaffected by its CTS_B signal. RTS_B remains asserted until the transfer is complete, even if the transmitter is disabled mid-way through a data transfer.

32.3.5.4 Transceiver driver enable using RTS_B

RS-485 is a multiple drop communication protocol in which the LPUART transceiver's driver is three-stated unless LPUART is driving. The transmitter can use the RTS_B signal to enable the driver of a transceiver. The polarity of RTS_B can be matched to the polarity of the transceiver's driver enable signal.

The following figure shows the receiver enable signal asserted. This connection can also connect RTS_B to both DE and RE_B. The transceiver's receiver is disabled when driving. A pullup can pull RXD to a nonfloating value during this time. You can refine this option further by operating LPUART in Single-Wire mode, freeing the RXD pin for other uses.

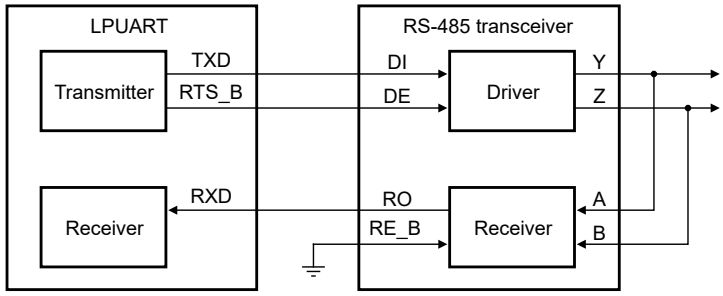


Figure 137. Transceiver driver enable using RTS_B

32.3.6 Receiver functional description

This section discusses the functioning of the LPUART receiver, as shown in the receiver portion of Block diagram. The section also discusses:

- The data sampling technique used to reconstruct receiver data.
- Different variations of the receiver wake-up function.

You can invert the receiver input by writing 1 to STAT[RXINV] and enable the receiver by writing 1 to CTRL[RE]. Character frames consist of a start bit of logic 0, along with N (7, 8, 9, 10) bits (MSB or LSB first), and one or two stop bits of logic 1. For information about 7-bit, 9-bit, or 10-bit Data mode, see Data modes. Going forward in this discussion, assume that LPUART is configured for a normal 8-bit Data mode.

After receiving the stop bit into the receive shifter, and provided the receive data register is not already full (STAT[RDRF] = 0), the data character is transferred to the receive FIFO, resulting in STAT[RDRF] becoming 1. However, if STAT[RDRF] is already 1, indicating that the receive data buffer is already full, STAT[OR] becomes 1 and the new data is lost.

Because the LPUART receiver is separate from the receive FIFO, the receive shift register can receive the next word when the receive FIFO is full, and it is only at the end of the character that the next data is written into the receive FIFO, potentially triggering the overrun flag if the FIFO is full.

When a program detects that the receive data register is full (STAT[RDRF] = 1), it gets the data from the FIFO by reading Data (DATA). See Interrupts for details about flag clearing.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

32.3.6.1 Data sampling technique

The LPUART receiver supports a configurable oversampling rate of between 4× and 32× of the baud rate clock for sampling. The receiver starts by considering logic level samples at the oversampling rate times the baud rate to search for a falling edge on the RXD serial data input pin. A falling edge is defined as a logic 0 sample after three consecutive logic 1 samples. The oversampling baud rate clock divides the bit time into 4 to 32 segments from 1 to OSR (where OSR is the configured oversampling ratio). When a falling edge is located, three more samples are taken at (OSR ÷ 2), (OSR ÷ 2) + 1, and (OSR ÷ 2) + 2 to ensure that this is a real start bit and not merely noise. If at least two of these three samples are 0, the receiver assumes they are synchronized to a received character. If another falling edge is detected before the receiver is considered synchronized, the receiver restarts sampling from the first segment.

The receiver then samples each bit time, including the start and stop bits, at (OSR ÷ 2), (OSR ÷ 2) + 1, and (OSR ÷ 2) + 2, to determine the logic level for that bit. The logic level is interpreted to be that of the majority of the samples taken during the bit time. If any sample in any bit time, including the start and stop bits, in a character frame fails to agree with the logic level for that bit, noise flag (STAT[NF]) becomes 1 when the received character is transferred to the receive FIFO.

When the LPUART receiver is configured to sample on both edges of the baud rate clock (that is, when BAUD[BOTHEDGE] = 1), the number of segments in each received bit is effectively doubled (from 1 to OSR× 2). The start and data bits are then sampled at OSR, OSR + 1, and OSR + 2. You must enable sampling on both edges of the clock for oversampling rates of 4× to 7×. This sampling is optional for higher oversampling rates.

The synchronization feature of LPUART synchronizes the internal oversampling counter with a detected falling edge on the receive signal, and to adjust the data sampling window. The falling edge detection needs three consecutive 1s prior to the "1->0" (one to zero) transition. After the initial falling edge detection for the start bit, the circuit continuously monitors the next falling edge, and resets the counter after another falling edge is detected. This synchronization to the start bit is termed as resynchronization.

When BAUD[RESYNCDIS] is 0, you perform this falling edge detection and resynchronization not only for the start bit but also for the rest of the character reception after the start bit.

When BAUD[RESYNCDIS] is 1, you perform the falling edge detection and resynchronization only for the start bit. The use case for disabling the resynchronization is protocols that require this (for example, LIN 2.1 prohibits resynchronization within a byte).

The following table and figure explain LPUART resynchronization.

Table 188. LPUART resynchronization settings

Resynchronization	BAUD[RESYNCDIS] = 0	BAUD[RESYNCDIS] = 1
For the starting bit falling edge	Yes	Yes
For all falling edges after the start bit	Yes	No

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

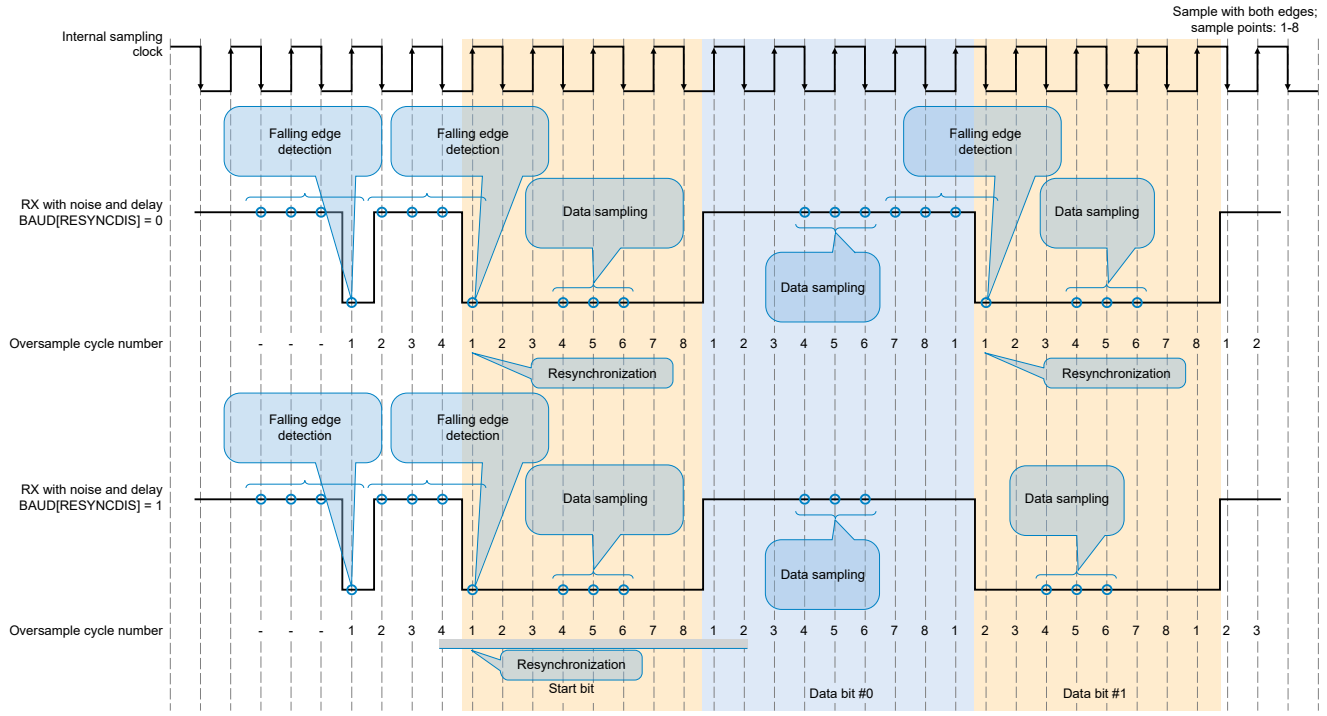


Figure 138. LPUART resynchronization diagram

32.3.6.1.1 Configurable sample point

In some non-standard UART signals where the valid bit period is only half of the actual bit period of a standard UART signal, chances are there that the receiver might misinterpret the bit value if it samples this incoming RX data packets at the middle of the bit period. The solution is to sample early the data so that correct bit value is received from the incoming RX data. To solve this issue LPUART has option to configure the sample point early. You need to configure CTRL[ERLSMP] = 1 to allow LPUART to sample the incoming data at 1/4 instant of the bit duration instead of the mid point.

32.3.6.2 Receiver wake-up operation

Receiver wake-up and receiver address matching are hardware mechanisms that allow an LPUART receiver to ignore the characters in a message intended for a different receiver.

During receiver wake-up, all receivers evaluate the first character(s) of each message, and as soon as they determine the message is intended for a different receiver, they write 1 to CTRL[RWU].

When CTRL[RWU] and STAT[RWUID] are 1, the status fields associated with the receiver, with the exception of STAT[IDLE], are inhibited from becoming 1, thus eliminating the software overhead for handling the unimportant message characters. At the end of a message, all receivers automatically force CTRL[RWU] to become 0. This results in all receivers waking up in time to look at the first character(s) of the next message.

During receiver address matching, the address matching is performed in hardware and the LPUART receiver ignores all characters that do not meet the address match requirements.

Table 189. Receiver wake-up options

CTRL[RWU]	BAUD[MAEN1] | BAUD[MAEN2]	BAUD[MATCFG]	CTRL[WAKE]: STAT[RWUID]	Receiver wake-up
0	0	X	X	Normal operation

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table 189. Receiver wake-up options (continued)

CTRL[RWU]	BAUD[MAEN1] | BAUD[MAEN2]	BAUD[MATCFG]	CTRL[WAKE]: STAT[RWUID]	Receiver wake-up
1	0	00	00	Receiver wake-up on idle line; STAT[IDLE] = 0
1	0	00	01	Receiver wake-up on idle line; STAT[IDLE] = 1
1	0	00	10	Receiver wake-up on address mark
1	1	11	10	Receiver wake-up on data match
0	1	00	X0	Address mark address match; STAT[IDLE] = 0 for discarded characters
0	1	00	X1	Address mark address match; STAT[IDLE] = 1 for discarded characters
0	1	01	X0	Idle line address match
0	1	10	X0	Match on and match off; STAT[IDLE] = 0 for discarded characters
0	1	10	X1	Match on and match off; STAT[IDLE] = 1 for discarded characters

32.3.6.2.1 Idle line wake-up

When CTRL[WAKE] is 0, you can configure the receiver for an idle line wake-up. In this mode, CTRL[RWU] becomes 0 automatically when the receiver detects a full character time of the idle-line level.

When CTRL[RWU] is set the receiver ignores everything until an idle condition is detected. After the idle condition is detected the CTRL[RWU] is cleared by hardware and the receiver starts receiving data.

The STAT[RWUID] indicates that whether the STAT[IDLE] flag is set or not as a result of the idle condition detection which clears CTRL[RWU].

The STAT[RWUID] says that if the STAT[IDLE] flag is to be set as a result of the idle condition which clears CTRL[RWU].

CTRL[M], CTRL[M7], and BAUD[M10] select 7-bit to 10-bit Data mode and BAUD[SBNS] selects a 1-bit or 2-bit stop bit number that determines how many bit times of idle are needed to constitute a full character time, 9 to 13 bit times because of the start and stop bits.

When CTRL[RWU] is 1 and STAT[RWUID] is 0, the idle condition that wakes up the receiver does not lead to STAT[IDLE] becoming 1. The receiver wakes up and waits for the first data character of the next message that leads to STAT[RDRF] becoming 1 and generates an interrupt if enabled. When STAT[RWUID] is 1, any idle condition leads to STAT[IDLE] becoming 1 and generates an interrupt if enabled, regardless of whether CTRL[RWU] is 0 or 1.

These are the ways to detect an idle line:

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

- When CTRL[ILT] is 0, the idle bit counter starts after the start bit so that the stop bit and any logic 1s at the end of a character count to calculate the full character time of idle.
- When CTRL[ILT] is 1, the idle bit counter does not start until after the stop bit time so that the data in the last character of the previous message does not impact the idle detection.

32.3.6.2.2 Address mark wake-up

When CTRL[WAKE] is 1, you can configure the receiver for an address mark wake-up. In this mode, CTRL[RWU] becomes 0 automatically when the receiver detects a logic 1 in the most significant bit of the received character. When parity is enabled, the second most significant bit is used for address mark wake-up.

Address mark wake-up allows messages to contain idle characters, but requires one bit to be reserved for use in address frames. The logic 1 in the most significant bit (or second most significant bit when parity is enabled) of an address frame writes 0 to CTRL[RWU] and writes 1 to STAT[RDRF]. In this case, the character with the address mark bit is received even if the receiver is sleeping during most of this character time.

32.3.6.2.3 Data match wake-up

When CTRL[RWU] and CTRL[WAKE] are 1, and BAUD[MATCFG] equals 11, the receiver is configured for a data match wake-up. In this mode, CTRL[RWU] becomes 0 automatically when the receiver detects a character that matches MATCH[MA1] when BAUD[MAEN1] is 1, or that matches MATCH[MA2] when BAUD[MAEN2] is 1.

32.3.6.2.4 Address match operation

You can enable the address match operation when either BAUD[MAEN1] or BAUD[MAEN2] is 1 and BAUD[MATCFG] is 0. In this function, a character that the RXD pin receives with a logic 1 in the most significant bit (or the second most significant bit when parity is enabled) is considered an address and is compared to the associated MATCH[MA1] or MATCH[MA2]. The character is only transferred to the receive buffer, and STAT[RDRF] becomes 1 if the comparison matches. All subsequent characters received with a logic 0 in the most significant bit (or the second most significant bit when parity is enabled) are considered to be data associated with the address and are transferred to the receive FIFO. If no marked address match occurs, no transfer is made to the receive FIFO, and all the characters that follow, with logic 0 in the most significant bit (or second most significant bit when parity is enabled), are also discarded. If both BAUD[MAEN1] and BAUD[MAEN2] are 0, the receiver operates normally, and all the received data is transferred to the receive FIFO.

The address match operation functions in the same way for both MATCH[MA1] and MATCH[MA2]:

- If either BAUD[MAEN1] or BAUD[MAEN2] is 1, a marked address is compared only to the associated Match Address (MATCH) and data is transferred to the receive FIFO only on a match.
- If both BAUD[MAEN1] and BAUD[MAEN2] are 1, a marked address is compared to both MATCH[MA1] and MATCH[MA2] and data is transferred only on a match with either of these fields.

32.3.6.2.5 Idle match operation

You can enable the idle match operation when either BAUD[MAEN1] or BAUD[MAEN2] is 1 and BAUD[MATCFG] is 1. In this function, the first character that the RXD pin receives after an idle line condition is considered an address and is compared to the associated MATCH[MA1] or MATCH[MA2]. The character is transferred only to the receive buffer, and STAT[RDRF] becomes 1, if the comparison matches. All subsequent characters are considered to be data associated with the address and are transferred to the receive FIFO until the next idle line condition is detected. If no address match occurs, no transfer is made to the receive FIFO, and all the frames that follow, until the next idle condition, are also discarded. If both BAUD[MAEN1] and BAUD[MAEN2] are 0, the receiver operates normally, and all the received data is transferred to the receive FIFO.

An idle match operation functions in the same way for both MATCH[MA1] and MATCH[MA2]:

- If either BAUD[MAEN1] or BAUD[MAEN2] is 1, the first character after an idle line is compared only to the associated Data (DATA) and data is transferred to the receive FIFO only on a match.
- If both BAUD[MAEN1] and BAUD[MAEN2] are 1, the first character after an idle line is compared to both MATCH[MA1] and MATCH[MA2] and data is transferred only on a match with either of these fields.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

32.3.6.2.6 Match on, match off operation

The match on, match off operation is enabled when both BAUD[MAEN1] and BAUD[MAEN2] are 1 and BAUD[MATCFG] = 10. In this function, a character that the RXD pin receives matches MATCH[MA1] and is transferred to the receive buffer, and STAT[RDRF] becomes 1. All subsequent characters are considered to be data and are also transferred to the receive FIFO, until a character that matches MATCH[MA2] is received. The character that matches MATCH[MA2], along with all subsequent characters, is discarded; and this continues until another character that matches MATCH[MA1] is received. If both BAUD[MAEN1] and BAUD[MAEN2] are 0, the receiver operates normally, and all the received data is transferred to the receive FIFO.

NOTE

The match on, match off operation requires both BAUD[MAEN1] and BAUD[MAEN2] to be 1.

32.3.6.3 Hardware flow control

To support hardware flow control, you can program the receiver to automatically assert and deassert RTS_B:

- RTS_B remains asserted until the transfer is complete, even if the transmitter is disabled midway through a data transfer. See Transceiver driver enable using RTS_B for more information.
- If the receiver RTS functionality is enabled, the receiver automatically deasserts RTS_B if STAT[RDRF] is 1 or a start bit is detected that causes STAT[RDRF] to become 1.
- The receiver asserts RTS_B when STAT[RDRF] is 0 and has not detected a start bit that causes STAT[RDRF] to become 1. There is no impact if STAT[RDRF] is 1 already.
- Even if RTS_B is deasserted, the receiver continues to receive characters until the receive FIFO is overrun.
- If the receiver RTS functionality is disabled, the receiver's RTS_B remains deasserted.
- When RTS is driven by receiver, it will be negated when the receiver is disabled irrespective of the FIFO level.

32.3.7 Additional LPUART functions

32.3.7.1 Data modes

You can configure the LPUART transmitter and receiver to operate in 7-bit Data mode by writing 1 to CTRL[M7], 9-bit Data mode by writing 1 to CTRL[M], or 10-bit Data mode by writing 1 to BAUD[M10]. In 9-bit Data mode, there exists a ninth data bit and in 10-bit mode, there exists a tenth data bit.

When performing 8-bit writes to the transmit FIFO, the ninth and tenth bits are pushed into the FIFO from CTRL[T8] and CTRL[T9]. For coherent 8-bit writes, you must write to CTRL[T8] and CTRL[T9] before writing to Data (DATA)[7:0]. However, if the values in CTRL[T8] or CTRL[T9] do not need to change, it is not necessary to update CTRL[T8] and CTRL[T9] before every 8-bit write to Data (DATA).

When performing 16-bit or 32-bit writes to the transmit FIFO, all 10 bits are pushed into the transmit FIFO from the write data.

When performing 8-bit reads of the receive FIFO, the ninth and tenth bits are held in CTRL[R8] and CTRL[R9] but you must read them before reading Data (DATA). A 16-bit or 32-bit read of the receive FIFO returns all 10 bits in Data (DATA).

The 9-bit Data mode is typically used with parity to allow eight bits of data plus the parity in the ninth bit, or it is used with the address mark wake-up so that the ninth data bit can serve as the wake-up bit. The 10-bit Data mode is typically used with parity and address mark wake-up so that the ninth data bit can serve as the wake-up bit and the tenth bit can serve as the parity bit. In custom protocols, the ninth and/or tenth bits can also serve as software-controlled markers.

32.3.7.2 Idle length

An idle character is one where the start bit, all data bits, and stop bits are in the mark position (idle state, generally logic 1). You can configure CTRL[ILT] to start detecting an idle character from the previous start bit (any data bits and stop bits count for idle character detection) or from the previous stop bit.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

You can also use CTRL[IDLECFG] to configure the number of idle characters that must be received before an idle line condition is detected. This field configures the number of idle characters that must be received before STAT[IDLE] becomes 1, STAT[RAF] becomes 0, and DATA[IDLINE] becomes 1 with the next received character.

CTRL[IDLECFG] also affects the idle line wake-up and idle match operations. When either the address match or match on/off operation is enabled, writing 1 to STAT[RWUID] causes any discarded characters to be treated as idle characters.

32.3.7.3 Loop mode

Enable Loop mode by setting CTRL[LOOPS] = 1 and CTRL[RSRC] = 0. You, sometimes, use Loop mode to check software, independent of connections in the external system, to help isolate system problems. In this mode, the transmitter output is internally connected to the receiver input and LPUART does not use the RXD pin.

Loop mode also internally connects the RTS_B output to the CTS_B input.

32.3.7.4 Single-Wire mode

Enable Single-Wire mode by setting CTRL[LOOPS] = 1 and CTRL[RSRC] = 1. Single-Wire mode implements a half-duplex serial connection. The receiver is internally connected to the transmitter output and TXD pin (the RXD pin is not used).

In Single-Wire mode, CTRL[TXDIR] controls the direction of serial data on the TXD pin. When CTRL[TXDIR] becomes 0, the TXD pin is an input to the receiver and the transmitter is temporarily disconnected from the TXD pin so that an external device can send serial data to the receiver. When CTRL[TXDIR] = 1, the TXD pin is an output that the transmitter drives. The internal loop back connection is disabled, and as a result, the receiver is unable to receive characters that the transmitter sends out.

32.3.8 Peripheral triggers

The connection of the LPUART peripheral triggers with other peripherals is chip-specific.

32.3.8.1 Output triggers

LPUART generates the following output triggers that can be connected to other peripherals on the chip:

- The transmit word trigger asserts at the end of each transmitted word and negates after 1-bit period.
- The transmit data trigger is identical to the TXD pin output, but without support for input trigger modulation.
- The receive word trigger asserts at the end of each received word that is written to the receive FIFO, for one oversampling clock period.
- The receive idle trigger asserts when STAT[IDLE] becomes 1, and negates when the next valid start bit is detected.

32.3.8.2 Input trigger

LPUART supports a peripheral input trigger that you can configure in one of the following ways:

- By enabling the CTS function: You can connect the input trigger (active high) instead of the CTS_B (active low) pin input. The input trigger must assert for longer than 1-bit clock period when the transmitter is idle, with data to send, to guarantee a new transmission.
- By making the input trigger modulate the transmit data output (trigger is logically ANDed with the TXD output): The input trigger is expected to be a free-running clock (carrier signal) that generates from a timer or PWM source with a frequency that is greater than the bit-clock frequency. The carrier signal must not toggle faster than the maximum supported bit time.
- By connecting the input trigger instead of the RXD pin input: The input trigger is expected to be generated from a receive data source, such as an analog comparator or external pin.

32.3.9 Infrared (IR) interface

LPUART provides the capability of transmitting narrow pulses to an IR LED and receiving narrow pulses, transforming them to serial bits, which are then sent to LPUART. The IrDA physical layer specification defines a half-duplex IR communication link for

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

exchanging data. The full standard includes data rates up to 16 Mbit/s. The LPUART IrDA support is limited to SIR mode that supports data rates only between 2.4 kbit/s and 115.2 kbit/s.

LPUART has an infrared transmit encoder and a receive decoder. The infrared decoder converts the received character from the IrDA format to the NRZ format, which the receiver uses. It also has an OSR oversampling baud rate clock counter that filters noise and indicates when a 1 is received. LPUART transmits serial bits of data, which the infrared submodule encodes, to transmit a narrow pulse for every zero bit. No pulse is transmitted for every single bit. When receiving data, an IR photo diode (external to LPUART) detects the IR pulses. The IR receive decoder transforms them to CMOS levels. The infrared receive decoder then stretches the narrow pulses to get back to a serial bit stream that LPUART receives. You can invert the polarity of transmitted pulses and expected receive pulses so that a direct connection can be made to external IrDA transceiver modules that use active-high pulses.

The IR submodule receives its clock sources from LPUART. The submodule selects one of these clocks to generate either 1 ÷ OSR, 2 ÷ OSR, 3 ÷ OSR, or 4 ÷ OSR narrow pulses during transmission.

32.3.9.1 Infrared transmit encoder

The infrared transmit encoder converts serial bits of data from the transmit shift register to the TXD signal. A narrow pulse is transmitted for a 0 bit and no pulse is transmitted for a 1 bit. The narrow pulse is sent at the start of the bit with a duration of 1 ÷ OSR, 2 ÷ OSR, 3 ÷ OSR, or 4 ÷ OSR of a bit time. A narrow low pulse is transmitted for a 0 bit when CTRL[TXINV] is 0, while a narrow high pulse is transmitted for a 0 bit when CTRL[TXINV] is 1.

32.3.9.2 Infrared receive decoder

The infrared receive block converts data from the RXD signal to the receive shift register. A narrow pulse is expected for each 0 received and no pulse is expected for each 1 received. A narrow low pulse is expected for a 0 bit when STAT[RXINV] is 0, while a narrow high pulse is expected for a 0 bit when STAT[RXINV] is 1. This receive decoder meets the edge jitter requirement as defined by the IrDA serial infrared physical layer specification.

32.3.9.3 Start-bit detection

When STAT[RXINV] is 0, the first falling edge of the received character corresponds to the start bit. The infrared decoder resets its counter. At this time, the receiver also begins its start bit detection process. After the start bit is detected, the receiver synchronizes its bit times to this start bit time. For the rest of the character reception, the infrared decoder's counter and the receiver's bit time counter count independently of each other.

32.3.9.4 Noise filtering

The decoder ignores any rising edges detected during the first half of the infrared decoder counter, and can leave any pulses less than one oversampling baud clock as undetected. This is regardless of whether the pulse is seen in the first or second half of the count.

32.3.9.5 Low-bit detection

During the second half of the decoder count, a rising edge is decoded as 0, which is sent to the receiver. The decoder counter is also reset.

32.3.9.6 High-bit detection

At OSR oversampling baud rate clocks after the previous rising edge, if a rising edge is not seen, the decoder sends a 1 to the receiver.

If the next bit is 0, which arrives late, a low bit is detected according to Low-bit detection. The value sent to the receiver is changed from 1 to 0. Then, if a noise pulse occurs outside the receiver's bit time sampling period, the delay of a 0 is not recorded as noise.

32.3.10 Modes of operation

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

32.3.10.1 Low-Power modes

LPUART remains functional during low-power modes, provided CTRL[DOZEEN] is 0 and the LPUART functional clock is enabled. LPUART can generate an interrupt or DMA request to cause a wake-up from low-power modes.

You can configure LPUART to be disabled in low-power modes, when CTRL[DOZEEN] is 1. In this case, the transmitter and receiver finish transmitting and receiving the current word.

If LPUART is disabled in low-power modes, it can generate a wake-up via STAT[RXEDGIF] if the receiver detects an active edge.

NOTE

See the chip-specific information for specific low-power modes available on your chip.

32.3.10.2 Debug mode

LPUART remains functional in Debug mode.

32.3.11 Clocking

Table 190. Types of clocks

Clock	Description
Functional	Is asynchronous to the bus clock and can remain enabled in Low-Power modes to support transmit and/or receive functions, including low-power wake-up.
Bus	Is only used for bus accesses to the control and configuration registers. The bus clock frequency must be sufficient to support the data bandwidth requirements of the LPUART transmit and receive registers, including the FIFOs.

32.3.12 Reset

Table 191. Types of resets

Reset	Description
Chip	Enables the logic and registers for the LPUART transmitter and receiver to reset to their default states.
Software	Resets the LPUART logic and registers to their default states, except for Global (GLOBAL). GLOBAL[RST] controls the LPUART software reset.
FIFO	Implements write-only control fields that reset the transmit FIFO (FIFO[TXFLUSH]) and receive FIFO (FIFO[RXFLUSH]). After a FIFO is reset, that FIFO becomes empty.

32.3.13 Interrupts

The LPUART transmitter has two status fields that can optionally generate hardware interrupt requests. If STAT[TDRE] is 1, it indicates that there is room in the transmit FIFO to write another transmit character to Data (DATA). If CTRL[TIE] is 1, a hardware interrupt is requested when STAT[TDRE] is 1.

STAT[TC] indicates that the transmitter is finished transmitting all data, preamble, and break characters and is idle with TXD at the inactive level. This field is often used in systems with modems to determine when it is safe to turn off the modem. If CTRL[TCIE] is 1, a hardware interrupt is requested when STAT[TC] is 1. Instead of hardware interrupts, software polling may be used to monitor STAT[TDRE] and STAT[TC] if the corresponding CTRL[TIE] or CTRL[TCIE] field is 0.

When a program detects that STAT[RDRF] is 1, it gets the data from this field by reading Data (DATA). The field becomes 0 by reading Data (DATA).

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

STAT[IDLE] includes logic that prevents it from becoming 1 repeatedly when the RXD line remains idle for an extended period of time. STAT[IDLE] becomes 0 when you write 1 to it, and cannot become 1 again until the receiver has received at least one new character and has 1 as the value of STAT[RDRF].

If the associated error is detected in the received character that caused STAT[RDRF] to become 1, STAT[NF], STAT[FE], and STAT[PF] become 1 at the same time STAT[RDRF] becomes 1. These flags do not become 1 in overrun cases.

If STAT[RDRF] is already 1 when a new character is ready to be transferred from the receive shifter to the receive FIFO, STAT[OR] becomes 1, instead of the data along with any associated STAT[NF], STAT[FE], or STAT[PF] condition getting lost.

If the received character matches the contents of MATCH[MA1] and/or MATCH[MA2], then STAT[MA1F] and/or STAT[MA2F] become 1 at the same time that STAT[RDRF] becomes 1.

At any time, an active edge on the RXD serial data input pin causes STAT[RXEDGIF] to become 1. STAT[RXEDGIF] becomes 0 when you write 1 to it. This function depends on the receiver being enabled (the value of CTRL[RE] being 1).

32.3.14 DMA

32.3.14.1 End-of-packet DMA transfers

The end-of-packet functionality is designed for serial interfaces where you may not know the size of the transfer in advance and the data is being pushed by an external device. The end-of-packet processing ensures that data does not become stranded in either the receive FIFO or the DMA receive buffer. Support for end-of-packet processing must be implemented in both the serial interfaces and DMA controller.

The condition that signals the end of packet is different for each serial interface, but the serial peripheral and DMA process it in the same way. For example, an idle line condition signals the UART end of packet, the Stop and/or Repeated Start condition signals the I2C end of packet, and PCS negation signals the SPI end of packet.

When the serial peripheral is configured to signal the end-of-packet condition to the DMA and the serial interface detects an end-of-packet condition, it asserts the DMA request for the receive FIFO irrespective of the watermark configuration. For larger watermark configurations, this ensures that the last few words of the transfer are first flushed from the receive FIFO.

The DMA then reads the contents of the receive FIFO, depending on the first word in the FIFO:

- If the receive FIFO is empty, the serial interface signals an end of packet condition to the DMA controller.
- If the receive FIFO is not empty, but the first word in the FIFO is the start of a new packet, data is not pulled from the receive FIFO and the serial interface signals an end-of-packet condition to the DMA controller.
- If the receive FIFO is not empty, and the first word in the FIFO is not the start of a new packet, the DMA transfers the receive data as normal.

Because the DMA may be transferring multiple words on each request, the end-of-packet condition persists until the DMA minor loop has completed and no additional data is pulled from the receive FIFO. The field that triggered the end-of-packet condition becomes 0 when the minor loop completes following the end of packet being signaled to the DMA controller.

When the DMA detects the end-of-packet condition, it writes all received words up to the end of packet into the system memory and saves the destination address for the word after the last valid data. The DMA then terminates the channel as if the major loop completed, including final offsets and optional interrupts, channel linking, and scatter-gather. You can, optionally, save the final destination address to the system memory. The final destination address is also available in the destination address register.

Because the DMA terminates the major loop, no servicing of the receive FIFO occurs until you or the hardware reconfigures the DMA (for example, channel linking or scatter-gather). You must minimize this delay to avoid receiver FIFO overrun. The automatic DMA end-of-packet processing is not recommended when there are only a few words transferred between end-of-packet conditions. This is because the DMA spends more time processing the end of packet than transferring the data. For example, the UART idle line length must be increased as needed to avoid an excessive number of idle conditions.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

32.4 External signals

Table 192. External signals

Signal	Description	I/O
TXD	Transmit data: This pin is normally an output, but is an input (tristated) in Single-Wire mode whenever the transmitter is disabled or the transmit direction is configured for receive data.	I/O
RXD	Receive data	I
CTS_B	Clear-to-send	I
RTS_B	Request-to-send	O

32.5 Initialization

Follow the below steps to send an UART frame (without DMA and hardware flow control):

1. After any reset
2. Do the following initialization sequence:
    a. Optional: configure CTRL[DOZEEN] = 0. This turns off any low power request through from system side
    b. Configure the targeted baud by writing appropriate values in BAUD[SBR], BAUD[OSR] (see Support of 32.768 KHz asynchronous module clock for baud calculation)
        i. Configure CTRL[MSEL32K]=1, GLOBAL[BSEL32K], BAUD[BOTHEDGE]=1, additionally if your baud clock is 32.768 KHz
    c. Configure the data mode, parity, stop bit, and bit polarity configuration by programming BAUD[M10]/ CTRL[M]/ CTRL[M7], CTRL[PE] and CTRL[PT], BAUD[SBNS], CTRL[TXINV]
    d. Configure interrupts: CTRL[TIE]/ CTRL[TCIE] / FIFO[TXOFE]
    e. Configure PINCFG[TRGSEL], CTRL[LOOPS], CTRL[RSRC], CTRL[TXDIR]
3. Configure TX FIFO related fields: FIFO[TXFE], WATER[TXWATER]
4. Turn on the transmitter CTRL[TE] = 1
5. Write DATA[T9/T8/T7/T6:T0]
6. Wait for the TX interrupt or poll the STAT[TC] to confirm the completion of the transmission
7. Repeat steps 5 – 6, for as many time equal to the no of data bytes
8. Disable the transmitter if you want to stop transmitting data
9. If you want to change the baud rate disable the transmitter gracefully (no active transactions happening) and follow step 2 - 4 again

Follow the below steps to receive an UART frame (without DMA and hardware flow control):

1. After any reset
2. Do the following initialization sequence:
    a. Optional: configure CTRL[DOZEEN] = 0. This turns off any low power request through from system side
    b. Configure the targeted baud by writing appropriate values in BAUD[SBR], BAUD[OSR] (see Support of 32.768 KHz asynchronous module clock for baud calculation)

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

i. Configure CTRL[MSEL32K]=1, GLOBAL[BSEL32K], BAUD[BOTHEDGE]=1, additionally if your baud clock is 32.768 KHz

c. Configure the data mode, parity, stop bit, bit polarity configuration by programming BAUD[M10]/ CTRL[M]/ CTRL[M7], CTRL[PE] and CTRL[PT], BAUD[SBNS], CTRL[TXINV]

d. Configure interrupts: CTRL[RIE]/ FIFO[RXUFE]

e. Configure PINCFG[TRGSEL], CTRL[LOOPS], CTRL[RSRC], STAT[AME]=0

3. Configure RX FIFO related fields: FIFO[RXFE], WATER[TXWATER]
4. Turn on the transmitter CTRL[RE] = 1
5. Send one frame from out side of LPUART and wait for the RX interrupt or poll the STAT[RAF] to confirm the completion of the reception
6. Check STAT[NF], STAT[PE], STAT[FE] for any noise-error, parity-error, or frame-error for the received frame
7. Repeat steps 5 – 7, for as many time equal to the no of data bytes
8. Disable the transmitter if you want to stop receiving data
9. If you want to change the baud rate disable the receiver gracefully (no active transactions happening) and follow step 2 - 4 again

32.6 LPUART register descriptions

LPUART includes registers to control baud rate, select options, report status, and store transmit and receive data. Access to an address outside the valid memory map generates a bus error.

NOTE

Writing to a read-only (RO) register or reading a write-only (WO) register can cause bus errors. LPUART does not verify whether programmed values in the registers are correct; you must write valid values to them.

32.6.1 LPUART memory map

LPUART0 base address: 4009_F000h

LPUART1 base address: 400A_0000h

LPUART2 base address: 400A_1000h

LPUART3 base address: 400A_2000h

Offset	Register	Width (In bits)	Access	Reset value
0h	Version ID (VERID)	32	R	0406_0003h
4h	Parameter (PARAM)	32	R	0000_0202h
8h	Global (GLOBAL)	32	RW	0000_0000h
Ch	Pin Configuration (PINCFG)	32	RW	0000_0000h
10h	Baud Rate (BAUD)	32	RW	0F00_0004h
14h	Status (STAT)	32	RW	00C0_0000h
18h	Control (CTRL)	32	RW	0000_0000h
1Ch	Data (DATA)	32	RW	0000_1000h

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Offset	Register	Width (In bits)	Access	Reset value
20h	Match Address (MATCH)	32	RW	0000_0000h
24h	MODEM IrDA (MODIR)	32	RW	0000_0000h
28h	FIFO (FIFO)	32	RW	00C0_0011h
2Ch	Watermark (WATER)	32	RW	0000_0000h
30h	Data Read-Only (DATARO)	32	R	0000_1000h

32.6.2 Version ID (VERID)

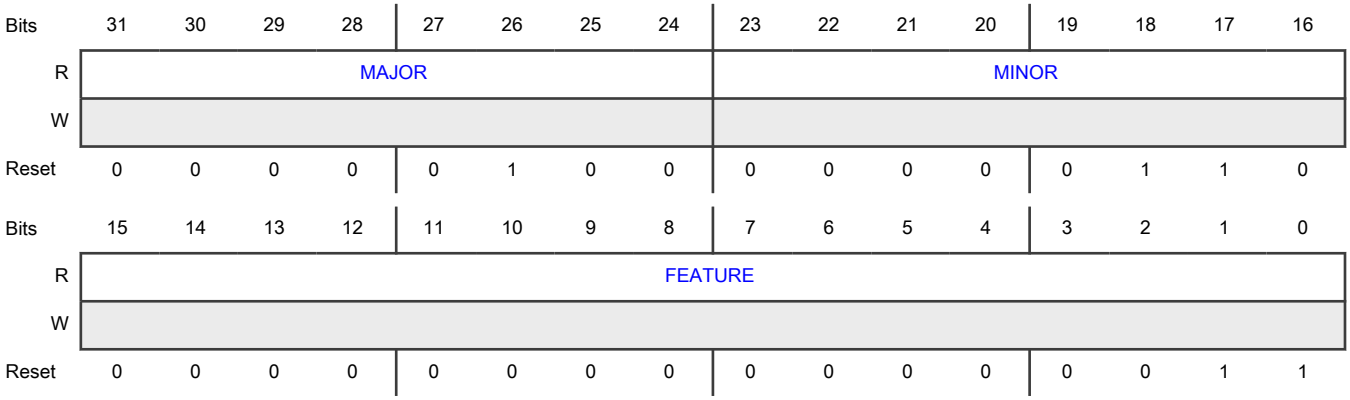
Offset

Register	Offset
VERID	0h

Function

Indicates the version integrated for this instance on the chip and also specifies the inclusion and exclusion of several optional features.

Diagram



Fields

Field	Function
31-24 MAJOR	Major Version Number Indicates the major version number for the module specification.
23-16	Minor Version Number

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
MINOR	Indicates the minor version number for the module specification.
15-0 FEATURE	Feature Identification Number Indicates the feature set number. 0000_0000_0000_0001b - Standard feature set 0000_0000_0000_0011b - Standard feature set with MODEM and IrDA support

32.6.3 Parameter (PARAM)

Offset

Register	Offset
PARAM	4h

Function

Indicates the parameter configuration for this instance on the chip.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RXFIFO								TXFIFO							
W																
Reset	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0

Fields

Field	Function
31-16 —	Reserved
15-8 RXFIFO	Receive FIFO Size Indicates an index used to determine the maximum capacity of the receive FIFO in terms of data units. • 1 data unit = a character plus any additional bits.

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	• The actual capacity is calculated as: — Capacity = 2 ^{RXFIFO} • Example: — If the RXFIFO field is 3, then the maximum capacity is: 2^3 = 8 data units
7-0 TXFIFO	Transmit FIFO Size Indicates an index used to determine the maximum capacity of the transmit FIFO in terms of data units. • 1 data unit = a character plus any additional bits. • The actual capacity is calculated as: — Capacity = 2 ^{TXFIFO} • Example: — If the TXFIFO field is 3, then the maximum capacity is: 2^3 = 8 data units

32.6.4 Global (GLOBAL)

Offset

Register	Offset
GLOBAL	8h

Function

Performs global functions.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0			0	0			STPFIX	0			BSEL32K	0		RST	0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Fields

Field	Function
31-13 —	Reserved
12 —	Reserved
11-9 —	Reserved
8 STPFIX	Fixed Stop Bit Configuration Use this field to program two stop bit in transmitting frame and one stop bit expectation in incoming frame. **NOTE** BAUD[SBNS] has an effect only when this field is 0. 0b - Normal stop bit configuration as per SBNS (default setting) 1b - Configures two stop bits in the transmitter and one stop bit in the receiver
7-5 —	Reserved
4 BSEL32K	Baud Select in 32.768 KHz Mode Use this field to select either of 9600 or 4800 baud generation using input 32.768 KHz. **NOTE** This field has effect only when BAUD[MSEL32K] = 1. 0b - Selects 9600 baud in 32.768 KHz mode (default setting) 1b - Selects 4800 baud in 32.768 KHz mode
3-2 —	Reserved
1 RST	Software Reset Specifies whether the module is reset. This field resets all internal logic and registers, except Global (GLOBAL). The reset takes effect immediately and remains asserted until you negate it. There is no minimum delay required before clearing the software reset. 0b - Not reset 1b - Reset
0	Reserved

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
—	

32.6.5 Pin Configuration (PINCFG)

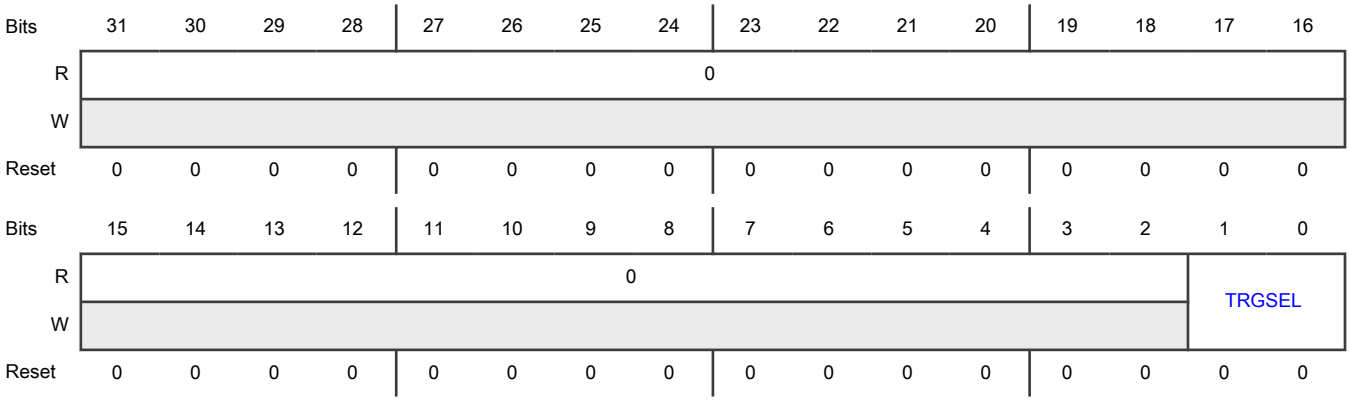
Offset

Register	Offset
PINCFG	Ch

Function

Enables the selection of input pins.

Diagram



Fields

Field	Function
31-2 —	Reserved
1-0 TRGSEL	Trigger Select Configures the input trigger usage. You must change the value of this field only when both the transmitter and receiver are disabled. 00b - Input trigger disabled 01b - Input trigger used instead of the RXD pin input 10b - Input trigger used instead of the CTS_B pin input 11b - Input trigger used to modulate the TXD pin output, which (after TXINV configuration) is internally ANDed with the input trigger

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

32.6.6 Baud Rate (BAUD)

Offset

Register	Offset
BAUD	10h

Function

Configures the baud rate.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	MAEN1	MAEN2	M10	OSR					TDMAE	MSEL32K	RDMAE	RIDMAE	MATCFG		BOTHEDGE	RESYNCD...
Reset	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	LBKDIE	RXEDGIE	SBNS	SBR												
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0

Fields

Field	Function
31 MAEN1	Match Address Mode Enable 1 Enables automatic address matching or data matching mode for MATCH[MA1]. If this field is 0, normal operation takes place. 0b - Disable 1b - Enable
30 MAEN2	Match Address Mode Enable 2 Enables automatic address matching or data matching mode for MATCH[MA2]. If this field is 0, normal operation takes place. 0b - Disable 1b - Enable
29 M10	10-Bit Mode Select Causes the tenth bit to be a part of the serial transmission. You must change the value of this field only when both the transmitter and receiver are disabled. 0b - Receiver and transmitter use 7-bit to 9-bit data characters 1b - Receiver and transmitter use 10-bit data characters

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
28-24 OSR	Oversampling Ratio Configures the OSR of the receiver. You must change the value of this field only when both the transmitter and receiver are disabled. **NOTE** BAUD[OSR] results in an OSR of BAUD[OSR] + 1, for example, BAUD[OSR] = 0_0101b results in a final division by 6. 0_0000b - Results in an OSR of 16 0_0001b - Reserved 0_0010b - Reserved 0_0011b - Results in an OSR of 4 (requires BAUD[BOTHEDGE] to be 1) 0_0100b - Results in an OSR of 5 (requires BAUD[BOTHEDGE] to be 1) 0_0101b - Results in an OSR of 6 (requires BAUD[BOTHEDGE] to be 1) 0_0110b - Results in an OSR of 7 (requires BAUD[BOTHEDGE] to be 1) 0_0111b - Results in an OSR of 8 0_1000b - Results in an OSR of 9 0_1001b - Results in an OSR of 10 0_1010b - Results in an OSR of 11 0_1011b - Results in an OSR of 12 0_1100b - Results in an OSR of 13 0_1101b - Results in an OSR of 14 0_1110b - Results in an OSR of 15 0_1111b - Results in an OSR of 16 1_0000b - Results in an OSR of 17 1_0001b - Results in an OSR of 18 1_0010b - Results in an OSR of 19 1_0011b - Results in an OSR of 20 1_0100b - Results in an OSR of 21 1_0101b - Results in an OSR of 22 1_0110b - Results in an OSR of 23 1_0111b - Results in an OSR of 24 1_1000b - Results in an OSR of 25 1_1001b - Results in an OSR of 26 1_1010b - Results in an OSR of 27

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	1_1011b - Results in an OSR of 28 1_1100b - Results in an OSR of 29 1_1101b - Results in an OSR of 30 1_1110b - Results in an OSR of 31 1_1111b - Results in an OSR of 32
23 TDMAE	Transmitter DMA Enable Enables STAT[TDRE] to generate a DMA request. 0b - Disable 1b - Enable
22 MSEL32K	32.768 KHz Mode Select Using this field, you can use a 32.786 KHz input clock to generate bauds as per GLOBAL[BSEL32K]. **NOTE** You need to disable resynchronization feature (BAUD[RESYNCDIS] = 1) when this field is 1. 0b - Selects normal baud generation (default setting) 1b - Selects special baud generation using 32.768 KHz input asynchronous clock
21 RDMAE	Receiver Full DMA Enable Enables STAT[RDRF] to generate a DMA request. 0b - Disable 1b - Enable
20 RIDMAE	Receiver Idle DMA Enable Enables STAT[IDLE] to generate a DMA request. If this field is 1, reading Data (DATA) when either DATA[RXEMPT] or DATA[IDLINE] is 1 generates an end-of-packet response until the completion of the DMA minor loop. During an end-of-packet response, reading Data (DATA) returns 0000_33FFh and does not pull data from the receive FIFO. STAT[IDLE] becomes 0 on completion of the minor loop, provided an end-of-packet response is generated and either the receive FIFO is empty or the receiver is active. 0b - Disable 1b - Enable
19-18 MATCFG	Match Configuration Configures the match addressing mode used. You must change the value of this field only when both the transmitter and receiver are disabled. 00b - Address match wake-up

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	01b - Idle match wake-up 10b - Match on and match off 11b - Enables RWU on data match and match on or off for the transmitter CTS input
17 BOTHEDGE	Both Edge Sampling Enables sampling of the received data on both edges of the baud rate clock, effectively doubling the number of times the receiver samples the input data for a given OSR. This field must be 1 for OSRs between x4 and x7 and is optional for higher OSRs. You must change the value of this field only when the receiver is disabled. If this field is 0, the receiver samples input data using the rising edge of the baud rate clock. If this field is 1, the receiver samples input data using the rising and falling edges of the baud rate clock. 0b - Rising edge 1b - Both rising and falling edges
16 RESYNCDIS	Resynchronization Disable Disables resynchronization of the received data word when a data one followed by data zero transition is detected. You must change the value of this field only when the receiver is disabled. 0b - Enable 1b - Disable
15 LBKDIE	LIN Break Detect Interrupt Enable Enables STAT[LBKDIF] to generate hardware interrupt requests. If this field is 0, hardware interrupts from STAT[LBKDIF] (uses polling) are disabled. If this field is 1, hardware interrupts are requested when STAT[LBKDIF] is 1. 0b - Disable 1b - Enable
14 RXEDGIE	RX Input Active Edge Interrupt Enable Enables STAT[RXEDGIF] to generate interrupt requests. If this field is 0, hardware interrupts from STAT[RXEDGIF] are disabled. If this field is 1, hardware interrupts are requested when STAT[RXEDGIF] is 1. Changing the value of CTRL[LOOPS] or CTRL[RSRC] when this field (RXEDGIE) is 1 can cause STAT[RXEDGIF] to become 1. 0b - Disable 1b - Enable
13 SBNS	Stop Bit Number Select Determines whether data characters include one or two stop bits.

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	You must change the value of this field only when both the transmitter and receiver are disabled. 0b - One stop bit 1b - Two stop bits
12-0 SBR	Baud Rate Modulo Divisor Sets the modulo divide rate for the baud rate generator. • If SBR is 0, baud rate generator is disabled. • If SBR is 1–8191, baud rate = baud clock ÷ ((OSR + 1) × SBR). You must update the 13-bit baud rate setting [SBR12:SBR0] only when both the transmitter and receiver are disabled (both CTRL[RE] and CTRL[TE] are 0).

32.6.7 Status (STAT)

Offset

Register	Offset
STAT	14h

Function
Provides the module status.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	LBKDIF	RXEDGIF	MSBF	RXINV	RWUID	BRK13	LBKDE	RAF	TDRE	TC	RDRF	IDLE	OR	NF	FE	PF
W	W1C	W1C										W1C	W1C	W1C	W1C	W1C
Reset	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	MA1F	MA2F	0				0		0						AME	LBKFE
W	W1C	W1C														
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31 LBKDIF	LIN Break Detect Interrupt Flag Indicates whether a LIN break character is detected.

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	This field becomes 1 when the LIN break detect circuitry is enabled and a LIN break character is detected. **NOTE** This field behaves differently for register reads and writes. When reading 0b - Not detected 1b - Detected When writing 0b - No effect 1b - Clear the flag
30 RXEDGIF	RXD Pin Active Edge Interrupt Flag Indicates whether an active edge on the receive pin has occurred. This field becomes 1 whenever the receiver is enabled and an active edge (falling if STAT[RXINV] is 0; rising if STAT[RXINV] is 1) on the RXD pin occurs. **NOTE** This field behaves differently for register reads and writes. When reading 0b - Not occurred 1b - Occurred When writing 0b - No effect 1b - Clear the flag
29 MSBF	MSB First Specifies the first bit that is transmitted and received after the start bit. If this field is 0, LSB (bit 0) is the first bit transmitted after the start bit and the first bit received after the start bit is identified as bit 0 of the data entity. If this field is 1, MSB (identified as bit 9, bit 8, bit 7, or bit 6) is the first bit that is transmitted, after the start bit, depending on the settings of CTRL[M], CTRL[PE], and BAUD[M10]. Writing 1 to this field reverses the order of the bits that are transmitted and received on the wire. This field does not affect the polarity of the bits, the location of the parity bit, or the location of the start or stop bits. You must change the value of this field only when both the transmitter and receiver are disabled. 0b - LSB 1b - MSB
28 RXINV	Receive Data Inversion Specifies whether receive data is inverted.

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	Writing 1 to this field reverses the polarity of the received data input before storing into FIFO. You must change the value of this field only when the receiver is disabled. **NOTE** Writing 1 to this field inverts the RXD input for all cases: data bits, start and stop bits, break, and idle. 0b - Not inverted 1b - Inverted
27 RWUID	Receive Wake Up Idle Detect Controls, for CTRL[RWU] on idle character detection, whether the idle character that wakes up the receiver writes 1 to STAT[IDLE]. For address match wake-up, this field controls whether STAT[IDLE] = 1 when the address does not match. You must change the value of this field only when the receiver is disabled. If this field is 0, during the Receive Standby state (CTRL[RWU] = 1), STAT[IDLE] does not become 1 upon detection of an idle character. During address match wake-up, STAT[IDLE] does not become 1 when an address does not match. If this field is 1, during the Receive Standby state (CTRL[RWU] = 1), STAT[IDLE] becomes 1 upon detection of an idle character. During address match wake-up, STAT[IDLE] becomes 1 when an address does not match. 0b - STAT[IDLE] does not become 1 1b - STAT[IDLE] becomes 1
26 BRK13	Break Character Generation Length Selects the longer transmitted break character length. The state of this field does not affect the detection of a framing error. You must change the value of this field only when the transmitter is disabled. You can send a break character by writing 1 to CTRL[SBK], or by writing the transmit FIFO when DATA[FRETSC] is 1 and DATA[R9T9] is 0. 0b - 9 to 13 bit times 1b - 12 to 15 bit times
25 LBKDE	LIN Break Detection Enable Enables LIN break detection. If this field is 0, LIN break detect is disabled, and only a normal break character can be detected. If this field is 1, LIN break detect is enabled and the LIN break character is detected at a length of 11 bit times (if CTRL[M] is 0), 12 bit times (if CTRL[M] is 1), or 13 bit times (if BAUD[M10] is 1). This field selects a longer break character detection length. When the field is 1, receive data is not stored in the receive FIFO.

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	NOTE This field enables the LIN break detect circuit and disables writing receive data to FIFO. Therefore, it ignores all characters except a LIN break. 0b - Disable 1b - Enable
24 RAF	Receiver Active Flag Indicates whether the LPUART receiver is idle or active. This field becomes 1 when the receiver detects the beginning of a valid start bit, and the field becomes 0 automatically when the receiver detects an idle line. 0b - Idle, waiting for a start bit 1b - Receiver active (RXD pin input not idle)
23 TDRE	Transmit Data Register Empty Flag Indicates whether the transmit FIFO level is greater than, equal to, or less than the watermark. After the transmit FIFO is enabled, this field becomes 1 when the number of datawords in the transmit FIFO is equal to, or less than the number that WATER[TXWATER] indicates. To make the value of this field 0, write to it until the number of words in the transmit FIFO is greater than the number that WATER[TXWATER] indicates. After the transmit FIFO is disabled, this field becomes 1 to indicate that the FIFO level is less than the watermark. To make the value of this field 0 again, write to Data (DATA). This register is not affected by a character that is in the process of being transmitted; it is updated at the start of each transmitted character. 0b - Greater than watermark 1b - Equal to or less than watermark
22 TC	Transmission Complete Flag Indicates whether the transmitter is active. This field becomes 0 when a transmission is in progress or a preamble or break character is loaded; in other words, when the transmitter is active (sending data, a preamble, or a break). The field becomes 1 when the transmit buffer is empty and no data, preamble, or break character is being transmitted; in other words, when the transmission activity is complete. When this happens, the transmit data output signal becomes idle (logic 1). This field becomes 0 after you write to Data (DATA) to transmit new data, queuing a preamble by first writing 0 and then writing 1 to CTRL[TE], queuing a break character by writing 1 to CTRL[SBK]. 0b - Transmitter active 1b - Transmitter idle
21 RDRF	Receive Data Register Full Flag Indicates whether the receive FIFO level is less than, equal to, or greater than the watermark. This field becomes 1 when the number of datawords in the receive buffer is greater than the number that WATER[RXWATER] indicates and the receive FIFO is enabled. To write 0 to this field, read Data

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	(DATA) until the number of datawords in the receive FIFO is equal to, or less than the number that WATER[RXWATER] indicates. When the receive FIFO is disabled, this field (RDRF) becomes 1 if the receive buffer (Data (DATA)) is full. To make this field 0, read Data (DATA). A character that is in the process of being received does not cause a change in this field until the entire character is received. Even if this field is 1, the character continues to be received until an overrun condition occurs after the entire character is received. 0b - Equal to or less than watermark 1b - Greater than watermark
20 IDLE	Idle Line Flag Indicates whether an idle line is detected. This field becomes 1 when the LPUART receive line becomes idle for a number of full character time defined by CTRL[IDLECFG] field after a period of activity. There are two conditions for idle detection: 1. When CTRL[ILT] is 0, the receiver starts counting idle bit times after the start bit. If the receive character is all 1s, these bit times and the stop bit time count towards the full character time of logic high, 10 to 13 bit times, needed for the receiver to detect an idle line. 2. When CTRL[ILT] is 1, the receiver does not start counting idle bit times until after the stop bits. **NOTE** The stop bits and any logic-high periods following the previous character do not count toward the required continuous logic-high duration that the receiver uses to detect an idle line. To clear this field (set it to 0), write a value of 1 to it. Once the field is cleared (set to 0), you cannot write 1 to it again until either: 1. A new character is stored in the receive buffer 2. A LIN break character sets STAT[LBKDIF] to 1 This field is set to 1 only once, even if the receive line remains idle for a long time. **NOTE** This field behaves differently for register reads and writes. When reading 0b - Idle line detected 1b - Idle line not detected When writing 0b - No effect 1b - Clear the flag
19 OR	Receiver Overrun Flag Indicates whether there is receive overrun.

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	This field becomes 1 when you cannot prevent STAT[RDRF] from overflowing with data. The field becomes 1 immediately after the stop bit is completely received for the dataword that overflows the buffer and all the other error fields (STAT[FE], STAT[NF], and STAT[PF]) are prevented from becoming 1. The data in the shift register is lost, but the data already in the LPUART data registers is not affected. If STAT[LBKDE] is enabled and a LIN break is detected, this field becomes 1 if STAT[LBKDIF] is not 0 before the next data character is received. When this field is 1, no additional data is stored in the receive FIFO even if sufficient room exists. **NOTE** This field behaves differently for register reads and writes. When reading 0b - No overrun 1b - Receive overrun (new LPUART data is lost) When writing 0b - No effect 1b - Clear the flag
18 NF	Noise Flag Indicates whether noise is detected in the received character of Data (DATA). The advanced sampling technique used in the receiver takes three samples in each of the received bits. If some of these samples disagree with the rest of the samples within any bit time in the frame, then noise is detected for that character. This field becomes 1 whenever the next character to be read from Data (DATA) is received with noise detected within the character. **NOTE** This field behaves differently for register reads and writes. When reading 0b - No noise detected 1b - Noise detected When writing 0b - No effect 1b - Clear the flag
17 FE	Framing Error Flag Indicates whether a framing error is detected. This field becomes 1 whenever the next character to be read from Data (DATA) is received with logic 0 detected where a stop bit was expected. **NOTE** This field behaves differently for register reads and writes.

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	When reading 0b - No framing error detected (this does not guarantee that the framing is correct) 1b - Framing error detected When writing 0b - No effect 1b - Clear the flag
16 PF	Parity Error Flag Indicates whether a parity error is detected. This field becomes 1 whenever the next character to be read from Data (DATA) is received when parity is enabled (CTRL[PE] is 1) and the parity bit in the received character does not agree with the expected parity value. **NOTE** This field behaves differently for register reads and writes. When reading 0b - No parity error detected 1b - Parity error detected When writing 0b - No effect 1b - Clear the flag
15 MA1F	Match 1 Flag Indicates whether the received data is equal to MATCH[MA1]. This field becomes 1 whenever the next character to be read from Data (DATA) matches the value of MATCH[MA1]. **NOTE** This field behaves differently for register reads and writes. When reading 0b - Not equal to MA1 1b - Equal to MA1 When writing 0b - No effect 1b - Clear the flag
14 MA2F	Match 2 Flag Indicates whether the received data is equal to MATCH[MA2].

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	This field becomes 1 whenever the next character to be read from Data (DATA) matches the value of MATCH[MA2]. **NOTE** This field behaves differently for register reads and writes. When reading 0b - Not equal to MA2 1b - Equal to MA2 When writing 0b - No effect 1b - Clear the flag
13-10 —	Reserved
9-8 —	Reserved
7-2 —	Reserved
1 AME	Address Mark Enable Configures the location of the address mark when configured for MSB first transfers. This field has no effect when configured for LSB first and you must change the value of this field only when both the transmitter and receiver are disabled. If this field is 0, address mark in character is MSB. If this field is 1, the address mark is stored in Data (DATA) at MSB (or MSB-1 when the parity bit is enabled). In other words, the address mark in character is the last bit before the stop bit (or parity bit when enabled). 0b - Disable 1b - Enable
0 LBKFE	LIN Break Flag Enable Enables the LIN break flag to assert whenever a LIN break character is detected. Unlike STAT[LBKDE], this does not impact data being stored in the receive data buffer, but does cause STAT[LBKDIF] to become 1 whenever a LIN break is detected. Because a LIN break is longer than a normal character, the LIN break triggers a write to STAT[RDRF] with the data fields as 0 and STAT[FE] as 1. The character following the LIN break has DATA[LINBRK] as 1 to indicate that the previous character was a LIN break. You must change the value of this field only when both the transmitter and receiver are disabled. If this field is 1, the LIN break character is detected at a length of 11-bit times (if CTRL[M] is 0), 12 (if CTRL[M] is 1), or 13 (if BAUD[M10] is 1).

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	0b - Disable 1b - Enable

32.6.8 Control (CTRL)

Offset

Register	Offset
CTRL	18h

Function

Controls various optional features of the LPUART system.

You must write to the fields of this register only when both the transmitter and receiver are disabled.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	R8T9	R9T8	TXDIR	TXINV	ORIE	NEIE	FEIE	PEIE	TIE	TCIE	RIE	ILIE	TE	RE	RWU	SBK
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MA1IE	MA2IE	ERLSMP	SWAP	M7	IDLECFG			LOOPS	DOZEEN	RSRC	M	WAKE	ILT	PE	PT
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31 R8T9	Receive Bit 8 Transmit Bit 9 Contains R8 and T9 that correspond to different functions. R8 is the ninth data bit received after you configure LPUART for 9-bit or 10-bit data formats. When reading 9-bit or 10-bit data, read R8 before reading Data (DATA). T9 is the tenth data bit transmitted after you configure LPUART for 10-bit data formats. When writing 10-bit data, write T9 before writing to Data (DATA). If T9 does not need to change from its previous value, such as when it is used to generate address mark or parity, then you need not write to it each time you write to Data (DATA).

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	NOTE R8 is a read-only bit and T9 is a write-only bit; the value read is different from the value written.
30 R9T8	Receive Bit 9 Transmit Bit 8 Contains R9 and T8 that correspond to different functions. R9 is the tenth data bit received after you configure LPUART for 10-bit data formats. When reading 10-bit data, read R9 before reading Data (DATA). T8 is the ninth data bit transmitted after you configure LPUART for 9-bit or 10-bit data formats. When writing 9-bit or 10-bit data, write T8 before writing to Data (DATA). If T8 does not need to change from its previous value, such as when it is used to generate address mark or parity, then you need not write to it each time you write to Data (DATA). **NOTE** R9 is a read-only field and T8 is a write-only field; the value read is different from the value written.
29 TXDIR	TXD Pin Direction in Single-Wire Mode Determines the direction of data at the TXD pin, in Single-Wire mode, when LPUART is configured for a single-wire half-duplex operation (CTRL[LOOPS] and CTRL[RSRC] are 1). When writing 0 to this field, the transmitter finishes transmitting the current character (if any) before the receiver starts receiving data from the TXD pin. 0b - Input 1b - Output
28 TXINV	Transmit Data Inversion Specifies whether transmit data is inverted. Writing 1 to this field reverses the polarity of the transmitted data output. This action inverts the TXD output for all cases: data bits, start and stop bits, break, and idle. 0b - Not inverted 1b - Inverted
27 ORIE	Overrun Interrupt Enable Enables STAT[OR] to generate hardware interrupt requests. When this field is 1, a hardware interrupt is requested. Use polling when OR interrupts are disabled. 0b - Disable 1b - Enable
26 NEIE	Noise Error Interrupt Enable Enables STAT[NF] to generate hardware interrupt requests. When this field is 1, a hardware interrupt is requested. Use polling when NF interrupts are disabled.

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	0b - Disable 1b - Enable
25 FEIE	Framing Error Interrupt Enable Enables STAT[FE] to generate hardware interrupt requests. When this field is 1, a hardware interrupt is requested. Use polling when FE interrupts are disabled. 0b - Disable 1b - Enable
24 PEIE	Parity Error Interrupt Enable Enables STAT[PF] to generate hardware interrupt requests. When this field is 1, a hardware interrupt is requested. Use polling when PF interrupts are disabled. 0b - Disable 1b - Enable
23 TIE	Transmit Interrupt Enable Enables STAT[TDRE] to generate interrupt requests if STAT[TDRE] is 1. 0b - Disable 1b - Enable
22 TCIE	Transmission Complete Interrupt Enable Enables STAT[TC] to generate interrupt requests if STAT[TC] is 1. 0b - Disable 1b - Enable
21 RIE	Receiver Interrupt Enable Enables STAT[RDRF] to generate hardware interrupt requests if STAT[RDRF] is 1. 0b - Disable 1b - Enable
20 ILIE	Idle Line Interrupt Enable Enables hardware interrupts. This field enables STAT[IDLE] to generate interrupt requests. If this field is 0, hardware interrupts from STAT[IDLE] are disabled and polling is used, and if this field is 1, hardware interrupts are enabled when STAT[IDLE] is 1. 0b - Disable 1b - Enable
19	Transmitter Enable

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
TE	Enables the LPUART transmitter. Using this field, you can also queue an idle preamble by first writing 0 and then writing 1 to this field. After this field becomes 0, the field reads 1 until the transmitter has completed the current character and the TXD pin is tristated. You can also queue a single idle character by writing to the transmit FIFO with DATA[FRETSC] and DATA[R9T9] = 1. 0b - Disable 1b - Enable
18 RE	Receiver Enable Enables the LPUART receiver. After you write 0 to this field, this field remains 1 until the receiver finishes receiving the current character (if any). 0b - Disable 1b - Enable
17 RWU	Receiver Wake-Up Control Specifies whether the LPUART receiver in standby is waiting for a wake-up condition. You can write 1 to this field to place the LPUART receiver in a Standby state. The field becomes 0 automatically when an RWU event occurs, that is, in case of an idle event when CTRL[WAKE] is 0 or an address match when CTRL[WAKE] is 1 and STAT[RWUID] is 0. **NOTE** You must write 1 to this field only when CTRL[WAKE] is 0 (wake-up on idle), if the channel is currently not idle. You can determine this by the value of STAT[RAF]. If the field is 1 to wake up an idle event and the channel is already idle, LPUART, possibly, discards the data. This is because the data must be received or a LIN break is detected after an Idle condition is detected before the IDLE flag is allowed to be reasserted. 0b - Normal receiver operation 1b - LPUART receiver in standby, waiting for a wake-up condition
16 SBK	Send Break Specifies whether queue break character(s) are to be sent. Writing 1 and then 0 to this field queues a break character in the transmit data stream. Additional break characters of 9 to 13 bits, or 12 to 15 bits if STAT[BRK13] is 1, and bit times of logic 0 are queued as long as this field is 1. Depending on the timing when this field is 1 and 0, relative to the character currently being transmitted, a second break character may be queued before you write 0 to this field. If the time taken to write 0 to this field is too long, for example, if the field does not become 0 by the end of the first break character, a second break character is sent. This is compared to queuing a break character through the transmit FIFO that guarantees only one break character is sent. You can also queue a single break character by writing to the transmit FIFO when DATA[FRETSC] is 1 and DATA[R9T9] is 0.

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	0b - Normal transmitter operation 1b - Queue break character(s) to be sent
15 MA1IE	Match 1 (MA1F) Interrupt Enable Enables the MA1F interrupt. 0b - Disable 1b - Enable
14 MA2IE	Match 2 (MA2F) Interrupt Enable Enables the MA2F interrupt. 0b - Disable 1b - Enable
13 ERLSMP	Early Sample Selection Using this field, you can configure sampling method of incoming asynchronous RX data. 0b - Regular midpoint bit sampling 1b - Early bit sampling at 1/2 duration of bit timing
12 SWAP	TXD and RXD Pin Swap Swaps TXD and RXD pins (set and cleared by software). If this field is 0, TX and RX are used in the standard way. If this field is 1, TX and RX are swapped, enabling an external UART module to work in a crossed-wired fashion. 0b - Use the standard way 1b - Swap
11 M7	7-Bit Mode Select Specifies the data characters that the receiver and transmitter use. You must change the value of this field only after both the transmitter and receiver are disabled. 0b - 8-bit to 10-bit 1b - 7-bit
10-8 IDLECFG	Idle Configuration Configures the number of continuous idle characters that must be received before an idle line condition is detected and STAT[IDLE] becomes 1, STAT[RAF] becomes 0, and DATA[IDLINE] becomes 1. 000b - 1 001b - 2 010b - 4 011b - 8

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	100b - 16 101b - 32 110b - 64 111b - 128
7 LOOPS	Loop Mode Select Selects Loop mode. After this field becomes 1, the RXD pin is disconnected from LPUART and the transmitter output is internally connected to the receiver input. The transmitter and receiver must be enabled to use the loop function. In Loop mode or Single-Wire mode, the transmitter outputs are internally connected to the receiver input (see CTRL[RSRC]). 0b - Normal operation: RXD and TXD use separate pins 1b - Loop mode or Single-Wire mode
6 DOZEEN	Doze Mode Enables LPUART in Doze mode. If this field is 1, LPUART remains active when not in Doze mode. 0b - Enable 1b - Disable
5 RSRC	Receiver Source Select Determines the source of the receiver shift register input if CTRL[LOOPS] is 1. This field has no effect unless CTRL[LOOPS] is 1. If this field is 0, internal Loopback mode is selected. LPUART does not use the RXD pin. Additionally, the CTS_B pin is not used and internally driven by the RTS_B output. If this field is 1, single-wire LPUART mode is selected where the TXD pin is connected to the transmitter output and receiver input. 0b - Internal Loopback mode 1b - Single-wire mode
4 M	9-Bit Or 8-Bit Mode Select Specifies the data characters that the receiver and transmitter use. 0b - 8-bit 1b - 9-bit
3 WAKE	Receiver Wake-Up Method Select Determines which condition wakes up LPUART when CTRL[RWU] = 1 and BAUD[MATCFG] = 0 (this field must be 1 when BAUD[MATCFG] = 11):

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	• Address mark in the bit preceding the stop bit (or bit preceding the parity bit when parity is enabled) of the received data character • An idle condition on the receive pin input signal If this field is 0, CTRL[RWU] is configured for idle line wake-up, and if this field is 1, CTRL[RWU] is configured with address mark wake-up. 0b - Idle 1b - Mark
2 ILT	Idle Line Type Select Determines when the receiver starts counting logic 1s as idle character bits. The count begins either after a valid start bit or the stop bit. If the count begins after the start bit, a string of logic 1s preceding the stop bit can cause false recognition of an idle character. Beginning the count after the stop bit avoids false idle character recognition, but requires properly synchronized transmissions. **NOTE** In case you write 1 to this field, a logic 0 is automatically shifted after a received stop bit, therefore resetting the idle count. 0b - After the start bit 1b - After the stop bit
1 PE	Parity Enable Enables hardware parity generation and checking. If parity is enabled, the bit immediately before the stop bit is treated as the parity bit. 0b - Disable 1b - Enable
0 PT	Parity Type Selects the type of parity, even or odd, if parity is enabled (CTRL[PE] = 1): • Odd parity means that the total number of logic 1 bits in the data character, including the parity bit, is odd. • Even parity means that the total number of 1s in the data character, including the parity bit, is even. 0b - Even parity 1b - Odd parity

32.6.9 Data (DATA)

Offset

Register	Offset
DATA	1Ch

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Function

Supports 8-bit, 16-bit, or 32-bit writes, each type of write performing a separate function. An 8-bit write to DATA[7:0] pushes {CTRL[R8T9], CTRL[R9T8], DATA[7:0]} the transmit FIFO with TSC clear. A 16-bit or 32-bit write pushes the data written into the FIFO and does not update the value of CTRL[R8T9] or CTRL[R9T8].

Reads and writes of this register are also involved in the automatic flag clearing mechanisms for some of the LPUART status fields.

NOTE

Reads return the contents of the read-only receive FIFO and writes go to the write-only transmit FIFO, making this register work as a set of two separate registers.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	NOISY	PARITYE	FRETSC	RXEMPT	IDLINE	LINBRK	R9T9	R8T8	R7T7	R6T6	R5T5	R4T4	R3T3	R2T2	R1T1	R0T0
W																
Reset	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-16 —	Reserved
15 NOISY	Noisy Data Received Indicates whether the current received dataword contained in DATA[R9:R0] is received with noise. 0b - Received without noise 1b - Received with noise
14 PARITYE	Parity Error Indicates whether the current received dataword contained in DATA[R9:R0] is received with a parity error. 0b - Received without a parity error 1b - Received with a parity error
13 FRETSC	Frame Error Transmit Special Character Specifies the way the dataword is received.

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	For reads, this field indicates that the current received dataword contained in DATA[R9:R0] is received with a frame error. For writes, the field indicates that a break or idle character is to be transmitted instead of the contents in DATA[T9:T0]. T9 indicates a break character when it is 0 and indicates an idle character when it is 1. The contents of DATA[T8:T0] must be 0. 0b - Received without a frame error on reads or transmits a normal character on writes 1b - Received with a frame error on reads or transmits an idle or break character on writes
12 RXEMPT	Receive Buffer Empty Indicates whether the receive buffer contains valid data. This field becomes 1 when there is no data in the receive buffer. The field does not consider data in the receive shift register. 0b - Valid data 1b - Invalid data and empty
11 IDLINE	Idle Line Indicates whether the receiver line was idle before receiving the character in DATA[9:0]. It can be read as “1” on the first character when the receiver is first enabled. The difference between this field and STAT[IDLE] is that, STAT[IDLE] flag does not set on an idle line after the receiver is first enabled, it needs to receive a character before it can become set, whereas this field does not have this limitation and can be set on the first character received if an idle line is detected beforehand. 0b - Not idle 1b - Idle
10 LINBRK	LIN Break Indicates whether the receiver line detected a LIN break before receiving the character in DATA[9:0]. This field requires the value of STAT[LBKDIF] to be 1. If this field is 0, the LIN break detect circuitry is disabled. 0b - Not detected 1b - Detected
9 R9T9	Read receive FIFO bit 9 or write transmit FIFO bit 9
8 R8T8	Read receive FIFO bit 8 or write transmit FIFO bit 8
7 R7T7	Read receive FIFO bit 7 or write transmit FIFO bit 7
6 R6T6	Read receive FIFO bit 6 or write transmit FIFO bit 6

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
5 R5T5	Read receive FIFO bit 5 or write transmit FIFO bit 5
4 R4T4	Read receive FIFO bit 4 or write transmit FIFO bit 4
3 R3T3	Read receive FIFO bit 3 or write transmit FIFO bit 3
2 R2T2	Read receive FIFO bit 2 or write transmit FIFO bit 2
1 R1T1	Read receive FIFO bit 1 or write transmit FIFO bit 1
0 R0T0	Read receive FIFO bit 0 or write transmit FIFO bit 0

32.6.10 Match Address (MATCH)

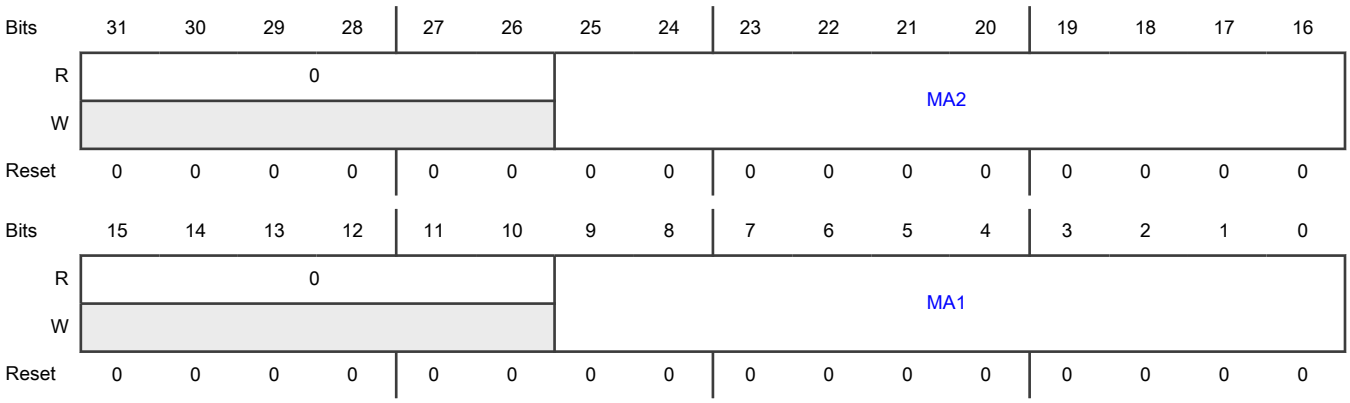
Offset

Register	Offset
MATCH	20h

Function

Provides addresses for address matching during the receiver operation.

Diagram



All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Fields

Field	Function
31-26 —	Reserved
25-16 MA2	Match Address 2 Is compared to input data addresses when the most significant bit is 1 and the associated Baud Rate (BAUD) field is 1. If a match occurs, the data that follows is transferred to Data (DATA). If a match fails, the data that follows is discarded. You must write to MATCH[MA1] and MATCH[MA2] only when the associated Baud Rate (BAUD) field is 0.
15-10 —	Reserved
9-0 MA1	Match Address 1 Is compared to input data addresses when the most significant bit is 1 and the associated Baud Rate (BAUD) field is 1. If a match occurs, the data that follows is transferred to Data (DATA). If a match fails, the data that follows is discarded. You must write to MATCH[MA1] and MATCH[MA2] fields only when the associated field in Baud Rate (BAUD) is 0.

32.6.11 MODEM IrDA (MODIR)

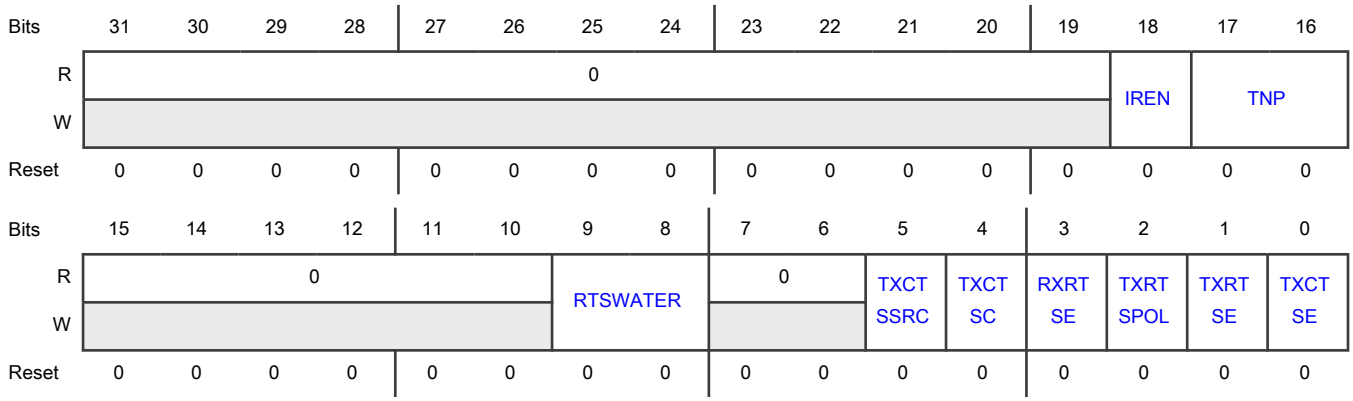
Offset

Register	Offset
MODIR	24h

Function

Controls options for setting the MODEM configuration.

Diagram



All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Fields

Field	Function
31-19 —	Reserved
18 IREN	IR Enable Enables IR modulation and demodulation. You must change the value of this field only when both the transmitter and receiver are disabled. 0b - Disable 1b - Enable
17-16 TNP	Transmitter Narrow Pulse Specifies whether LPUART transmits a 1 ÷ OSR, 2 ÷ OSR, 3 ÷ OSR, or 4 ÷ OSR narrow pulse when the IR pulse is enabled. You must change the value of this field only when both the transmitter and receiver are disabled. The IR pulse width must be configured to less than half of the OSR. Common pulse widths are 3 ÷ 16, 1 ÷ 16, 1 ÷ 32, or 1 ÷ 4 of the bit length. You can configure these by selecting the appropriate OSR and pulse width. 00b - 1 ÷ OSR 01b - 2 ÷ OSR 10b - 3 ÷ OSR 11b - 4 ÷ OSR
15-10 —	Reserved
9-8 RTSWATER	Receive RTS Configuration Configures the assertion and negation of the receiver's RTS_B output. The receiver's RTS_B output negates when the number of empty words in the receive FIFO is greater or equal to the value of this field. If this field is 0, the RTS_B pin negates when the receive FIFO is full. For the purpose of receive RTS_B generation, the number of words in the receive FIFO updates when a start bit is detected. This supports additional latency between RTS_B negation and the external transmitter ceasing transmission. If both receive RTS_B and address or data matching is enabled, RTS_B could assert at the end of a character if there exists no match. You must change the value of this field only when the receiver is disabled.
7-6 —	Reserved
5 TXCTSSRC	Transmit CTS Source Configures the source of the CTS input. 0b - The CTS_B pin 1b - An internal connection to the receiver address match result

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
4 TXCTSC	Transmit CTS Configuration Configures whether the CTS state or input is checked or sampled at the start of each character or only when the transmitter is idle. 0b - Sampled at the start of each character 1b - Sampled when the transmitter is idle
3 RXRTSE	Receiver RTS Enable Allows the RTS output to control the CTS input of the transmitting device to prevent receiver overrun. You must change the value of this field only when the receiver is disabled. If this field is 0, the receiver has no effect on RTS. If this field is 1, RTS is deasserted if STAT[RDRF] is 1 or a start bit is detected that causes STAT[RDRF] to become 1. RTS is asserted if STAT[RDRF] is 0 and has not detected a start bit that causes STAT[RDRF] to become 1. **NOTE** Do not write 1 to both MODIR[RXRTSE] and MODIR[TXRTSE]. 0b - Disable 1b - Enable
2 TXRTSPOL	Transmitter RTS Polarity Controls the polarity of the transmitter RTS. This field does not affect the polarity of the receiver RTS that remains negated in the active-low state unless MODIR[TXRTSE] is 1. You must change the value of this field only when the transmitter is disabled. 0b - Active low 1b - Active high
1 TXRTSE	Transmitter RTS Enable Controls the operation of RTS before and after a transmission. You must change the value of this field only when the transmitter is disabled. If this field is 0, the transmitter has no effect on RTS, and if this field is 1, a character is placed into an empty transmit shift register. RTS asserts 1-bit time before the start bit is transmitted and deasserts 1-bit time after all characters in the transmitter FIFO and shift register are completely sent, including the last stop bit. 0b - Disable 1b - Enable
0 TXCTSE	Transmitter CTS Enable Enables the operation of the transmitter. You can write 1 to this field irrespective of the states of MODIR[TXRTSE] and MODIR[RXRTSE]. If this field is 1, the transmitter checks the state of the CTS signal each time it is ready to send a character. If CTS is asserted, the character is sent. If CTS is deasserted, the TXD signal remains in the mark state and

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	transmission is delayed until CTS is asserted. Changes in CTS, when a character is being sent, do not affect its transmission. 0b - Disable 1b - Enable

32.6.12 FIFO (FIFO)

Offset

Register	Offset
FIFO	28h

Function

Provides you the ability to turn on and turn off the FIFO functionality.

This register also provides you the size of the FIFO that has been implemented. You can read this register at any time and must write to it only when CTRL[RE] and CTRL[TE] are 0 and the FIFO is empty.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0								TXEMPT	RXEMPT	0				TXOF	RXUF
W															W1C	W1C
Reset	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	RXIDEN			TXOFE	RXUFE	TXFE	TXFIFOSIZE			RXFE	RXFIFOSIZE		
W	TXFLUSH	RXFLUSH														
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1

Fields

Field	Function
31-24 —	Reserved
23 TXEMPT	Transmit FIFO Or Buffer Empty Indicates whether the transmit buffer is empty.

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	This field becomes 1 when there is no data in the transmit FIFO or buffer. The field does not consider data in the transmit shift register. 0b - Not empty 1b - Empty
22 RXEMPT	Receive FIFO Or Buffer Empty Indicates whether the receive buffer is empty. This field becomes 1 when there is no data in the receive FIFO or buffer. The field does not consider data in the receive shift register. 0b - Not empty 1b - Empty
21-18 —	Reserved
17 TXOF	Transmitter FIFO Overflow Flag Indicates whether more data has been written to the transmit FIFO than it can hold. If this field is 0, no transmit FIFO overflow has occurred since the last time the field was cleared, and if this field is 1, at least one transmit FIFO overflow has occurred since the last time the field was cleared. This field becomes 1 regardless of the value of FIFO[TXOFE]. However, an interrupt is issued to the host only if FIFO[TXOFE] is 1. **NOTE** This field behaves differently for register reads and writes. When reading 0b - No overflow 1b - Overflow When writing 0b - No effect 1b - Clear the flag
16 RXUF	Receiver FIFO Underflow Flag Indicates whether more data has been read from the receive FIFO than was present. If this field is 0, no receive FIFO underflow has occurred since the last time the field was cleared, and if this field is 1, at least one receive FIFO underflow has occurred since the last time the field was cleared. This field becomes 1 regardless of the value of FIFO[RXUFE]. However, an interrupt is issued to the host only if FIFO[RXUFE] is 1. **NOTE** After reset if DATA register at offset 1Ch is read without any data in FIFO it sets this field.

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	NOTE This field behaves differently for register reads and writes. When reading 0b - No underflow 1b - Underflow When writing 0b - No effect 1b - Clear the flag
15 TXFLUSH	Transmit FIFO Flush Causes all data that is stored in the transmit FIFO to be flushed. If you write 0 to this field, no flush operation occurs, and if you write 1 to this field, all data in the transmit FIFO or buffer clears out. This does not affect data in the transmit shift register. 0b - No effect 1b - All data flushed out
14 RXFLUSH	Receive FIFO Flush Causes all data that is stored in the receive FIFO to be flushed. If you write 0 to this field, no flush operation occurs, and if you write 1 to this field, all data in the receive FIFO or buffer clears out. This does not affect data in the receive shift register. 0b - No effect 1b - All data flushed out
13 —	Reserved
12-10 RXIDEN	Receiver Idle Empty Enable Enables STAT[RDRF] to become 1 when the receiver is idle for a number of idle characters and the FIFO is not empty. 000b - Disable STAT[RDRF] to become 1 because of partially filled FIFO when the receiver is idle 001b - Enable STAT[RDRF] to become 1 because of partially filled FIFO when the receiver is idle for one character 010b - Enable STAT[RDRF] to become 1 because of partially filled FIFO when the receiver is idle for two characters 011b - Enable STAT[RDRF] to become 1 because of partially filled FIFO when the receiver is idle for four characters

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	100b - Enable STAT[RDRF] to become 1 because of partially filled FIFO when the receiver is idle for eight characters 101b - Enable STAT[RDRF] to become 1 because of partially filled FIFO when the receiver is idle for 16 characters 110b - Enable STAT[RDRF] to become 1 because of partially filled FIFO when the receiver is idle for 32 characters 111b - Enable STAT[RDRF] to become 1 because of partially filled FIFO when the receiver is idle for 64 characters
9 TXOFE	Transmit FIFO Overflow Interrupt Enable Enables FIFO[TXOF] to generate an interrupt to the host. 0b - Disable 1b - Enable
8 RXUFE	Receive FIFO Underflow Interrupt Enable Enables FIFO[RXUF] to generate an interrupt to the host. 0b - Disable 1b - Enable
7 TXFE	Transmit FIFO Enable Enables the transmit FIFO. If this field is 0, the transmit buffer operates as a FIFO of depth equal to 1 dataword, regardless of the value in FIFO[TXFIFOSIZE]. Both CTRL[TE] and CTRL[RE] must be 0 before you change the value of this field. If this field is 1, the built-in FIFO structure for the transmit buffer is enabled. FIFO[TXFIFOSIZE] indicates the size of the FIFO structure. 0b - Disable 1b - Enable
6-4 TXFIFOSIZE	Transmit FIFO Buffer Depth Indicates the maximum number of transmit datawords (transmit FIFO buffer depth) that can be stored in the transmit buffer. 000b - 1 001b - 4 010b - 8 011b - 16 100b - 32 101b - 64 110b - 128

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	111b - 256
3 RXFE	Receive FIFO Enable Enables the receive FIFO. If this field is 0, the receive buffer operates as a FIFO of depth equal to 1 dataword, regardless of the value in FIFO[RXFIFOSIZE]. Both CTRL[RE] and CTRL[TE] must be 0 before you change the value of this field. If this field is 1, the built-in FIFO structure for the receive buffer is enabled. FIFO[RXFIFOSIZE] indicates the size of the FIFO structure. 0b - Disable 1b - Enable
2-0 RXFIFOSIZE	Receive FIFO Buffer Depth Indicates the maximum number of receive datawords (receive FIFO buffer depth) that can be stored in the receive buffer before an overrun occurs. 000b - 1 001b - 4 010b - 8 011b - 16 100b - 32 101b - 64 110b - 128 111b - 256

32.6.13 Watermark (WATER)

Offset

Register	Offset
WATER	2Ch

Function

Provides the ability to set a programmable threshold for notification, or sets the programmable thresholds to indicate that transmit data can be written or receive data can be read.

You may read this register at any time but must write to it only when CTRL[TE] is 0.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0					RXCOUNT			0						RXWATER	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0					TXCOUNT			0						TXWATER	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-27 —	Reserved
26-24 RXCOUNT	Receive Counter Indicates the number of datawords in the receive FIFO or buffer. If a dataword is being received in the receive shift register, it is not included in the count. This value may be used in conjunction with FIFO[RXFIFOSIZE] to calculate the room left in the receive FIFO or buffer.
23-18 —	Reserved
17-16 RXWATER	Receive Watermark Generates an interrupt or a DMA request if the number of datawords in the receive FIFO or buffer is greater than the value of this field. For proper operation, the value of this field must be less than the size of the receive FIFO or buffer, as indicated by FIFO[RXFIFOSIZE] and FIFO[RXFE].
15-11 —	Reserved
10-8 TXCOUNT	Transmit Counter Indicates the number of datawords in the transmit FIFO or buffer. If a dataword is being transmitted to the transmit shift register, it is not included in the count. This value may be used in conjunction with the value of FIFO[TXFIFOSIZE] to calculate the room left in the transmit FIFO or buffer.
7-2 —	Reserved
1-0	Transmit Watermark

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
TXWATER	Generates an interrupt or a DMA request when the number of datawords in the transmit FIFO or buffer is equal to or less than the value of this field. For proper operation, the value of this field must be less than the size of the transmit buffer or FIFO, as indicated by FIFO[TXFIFOSIZE] and FIFO[TXFE].

32.6.14 Data Read-Only (DATARO)

Offset

Register	Offset
DATARO	30h

Function

Indicates the first entry in the receive FIFO, but does not pull data from the FIFO.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DATA															
W																
Reset	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-16 —	Reserved
15-0 DATA	Receive Data Indicates the first entry from the receive FIFO. This register has the same functionality as that of Data (DATA).

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Chapter 33
Low Power Serial Peripheral Interface (LPSPI)

33.1 Chip-specific LPSPI information

Table 193. Reference links to related information

Topic	Related module	Reference
Full description	LPSPI	LPSPI
System memory map		Memory map
Clocking		Clock distribution
Power Management		Power Management
Signal multiplexing	Port control	Signal multiplexing

33.1.1 Module instances

This chip supports one instance of LPSPI: LPSPI0.

33.2 Overview

LPSPI provides an efficient interface (either as a controller or peripheral) to an SPI bus, which is a synchronous serial communication interface used in embedded systems. It is typically used to perform short distance communications between microcontrollers and peripheral devices, on printed circuit boards. Typical applications include interfacing with secure digital cards and LCD displays.

NOTE

The terminology in this chapter has been updated to align with NXP's inclusive language standards, as shown in the table below.

Table 194. Updated terms

Updated term	Deprecated term
Controller	Master
Peripheral	Slave

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

33.2.1 Block diagram

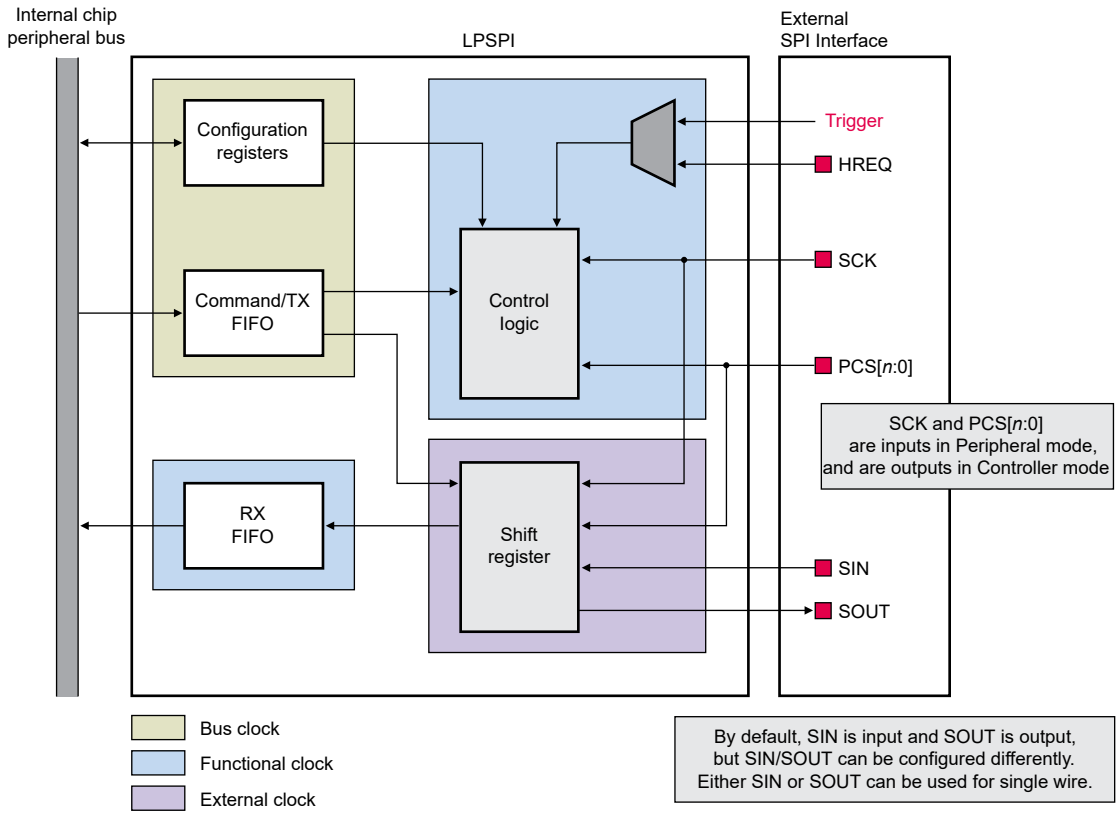


Figure 139. Block diagram

33.2.2 Features

- Minimal CPU overhead, with DMA transmit and receive requests supporting FIFO register accesses
- Operation continues in Deep Sleep mode, if configured to do so and an appropriate clock is available
- Support available for 32-bit word size
- Configurable clock polarity and phase
- Support available for 4 peripheral chip selects in Controller mode
- Support available for Peripheral mode
- 4-word transmit and command FIFO
- 4-word receive FIFO
- Flexible timing parameters in Controller mode, including SCK frequency and duty cycle, and delays between PCS and SCK edges
- Continuous transfer option to keep PCS asserted across multiple frames
- Full-duplex transfers that support 1-bit transmit and receive on each clock edge
- Half-duplex transfers that support:
  - 1-bit transmit or receive on each clock edge
  - 2-bit transmit or receive on each clock edge
  - 4-bit transmit or receive on each clock edge

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

- Option to use host request to control the start of an SPI bus transfer
- Receive data match logic that discards nonmatching data and interrupt on data match

33.3 Functional description

33.3.1 Controller mode

33.3.1.1 Transmit and command FIFO commands

The transmit and command FIFO is a combined FIFO that includes both transmit data words and command words. You store:

- Transmit data words in the transmit and command FIFO, by writing to Transmit Data (TDR).
- Command words in the transmit and command FIFO, by writing to Transmit Command (TCR).

When a command word is at the top of the transmit and command FIFO, the actions that can occur depend on whether LPSPI is busy or between frames (see TCR[CONT] and TCR[CONTC]). See Table 195 for conditions and possible corresponding actions when a command word is at the top of the transmit and command FIFO.

Table 195. Possible actions when a command word is at the top of the transmit and command FIFO

Condition	Action
LPSPI is enabled and idle.	The command word is pulled from the FIFO, and this command word controls all subsequent transfers.
LPSPI is busy and TCR[CONTC] is 0.	The SPI frame completes at the end of the existing word, ignoring TCR[FRAMESZ]. The command word is then pulled from the FIFO and that command word controls all subsequent transfers (or until the next update to the command word). Note that a command word with TCR[CONTC] = 0 always terminates the existing transfer regardless of the previous TCR[CONT] value.
LPSPI is busy; the existing TCR[CONT] value is 1 and the new TCR[CONTC] value is 1.	The command word must be updated at the frame boundary. The command word is pulled from the FIFO during the last SCK pulse of the existing frame (based on the value of FRAMESZ), and the frame continues using the new command value for the rest of the frame (or until the next update to the command word). When TCR[CONTC] = 1, only the lower 24 bits of the command word are updated. If the command word is updated at a word boundary, then the transfer halts (stops) after that word. TCR[CONTC] is ignored when not at a frame boundary, so the frame ends prematurely.

TCR[CONT] = 1 keeps PCS asserted at end of frame, allowing the transfer to continue.

TCR[CONTC] = 1 specifies that this command word must not terminate the existing frame, and the transfer can continue using the new command word.

TCR[CONTC] = 1 is restricted in the sense that the new command must load on a frame boundary, and the only way for a transfer to continue from a frame boundary is when the previous command has TCR[CONT] = 1.

You can read the current state of the existing command word from Transmit Command (TCR). It requires at least three LPSPI functional clock cycles for Transmit Command (TCR) to update after you write to it (assuming an empty FIFO), and LPSPI must be enabled (CR[MEN] = 1).

Writing to Transmit Command (TCR) does not initiate an SPI bus transfer, unless TCR[TXMSK] = 1. When TCR[TXMSK] = 1, a new command word is not loaded until the end of the existing frame (based on the value of TCR[FRAMESZ]); at the end of the transfer, TCR[TXMSK] transitions to 0.

In Controller mode, the LPSPI command word in Transmit Command (TCR) controls SPI attributes based on the selections in register fields. See Table 196 for TCR fields and associated functionality related to data transfer.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table 196. Command word in Controller mode

Transmit Command (TCR)		Description	Can this field be modified during a data transfer?
Field	Name		
CPOL	Clock polarity	Specifies the polarity of the SCK pin. Any change of CPOL value causes a transition on the SCK pin.	N
CPHA	Clock phase	Specifies the clock phase of the transfer.	N
PRESCALE	Prescaler value	Specifies a prescaler used to divide the LPSPI functional clock, to generate the timing parameters of the SPI bus transfer. Changing PRESCALE in conjunction with PCS enables LPSPI to connect to different peripheral devices at different frequencies.	N
PCS	Peripheral chip select	Specifies which PCS pin asserts for the transfer; the polarity of PCS is static and specified by CFGR1[PCSPOL]. If CFGR1[PCSCFG] = 1, do not select PCS[3:2].	N
LSBF	LSB first	Specifies whether LSB (bit 0) or MSB (bit 31 for a 32-bit word) is transmitted or received first.	Y
BYSW	Byte swap	Enables byte swap on each 32-bit word when transmitting and receiving data. Byte swapping can be useful when interfacing with devices that organize data as big-endian.	Y
CONT	Continuous transfer	Configures LPSPI for a continuous transfer that keeps PCS asserted between frames (as specified by FRAMESZ). You must write a new command word to cause PCS to negate. Also, this field supports changing the command word at frame size boundaries.	Y
CONTC	Continuing command	Indicates that this is a new command word for the existing continuous transfer. When CONTC = 1, the command word must only be written to the transmit and command FIFO on a frame boundary.	Y
RXMSK	Receive data mask	Masks the receive data and does not store the masked receive data in the receive FIFO or perform receive data matching. This option is useful for half-duplex transfers or to specify which fields are compared during receive data matching.	Y
TXMSK	Transmit data mask	Masks the transmit data; masked transmit data is not pulled from the transmit FIFO, and the output data pin is 3-stated (unless otherwise configured by CFGR1[OUTCFG]). This option is useful for half-duplex transfers.	Y
WIDTH	Transfer width	Specifies the number of bits shifted on each SCK pulse: • 1-bit transfers support traditional SPI bus transfers in either half-duplex or full-duplex data formats. • 2-bit and 4-bit half-duplex transfers are useful for interfacing with QuadSPI memory devices, and either TXMSK or RXMSK must also be 1.	Y
FRAMESZ	Frame size	Configures the frame size in number of bits equal to (FRAMESZ + 1):	Y

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table 196. Command word in Controller mode (continued)

Transmit Command (TCR)		Description	Can this field be modified during a data transfer?
Field	Name		
		• The minimum frame size is 8 bits. • If the frame size is larger than 32 bits, then the frame is divided into multiple words of 32 bits; each word is loaded from the transmit FIFO and stored in the receive FIFO separately. • If the size of the frame is not divisible by 32, then the last load of the transmit FIFO and store of the receive FIFO contains the remaining bits. For example, a 72-bit transfer consists of three words: the first and second words are 32 bits, and the third word is 8 bits.	

33.3.1.1.1 SPI bus transfers

LPSPI initiates an SPI bus transfer when all these conditions are true:

- Data is written to the transmit FIFO.
- The HREQ pin is asserted (or the HREQ function is disabled).
- LPSPI is enabled.

To perform the SPI bus transfer, LPSPI uses the attributes configured in Transmit Command (TCR) and the timing parameters defined in Clock Configuration (CCR).

The SPI bus transfer ends after the number of bits indicated by the value of FRAMESZ have been transferred (provided CONT = 0), or at the end of a word when a new transmit command word is at the top of the transmit and command FIFO. When LPSPI is disabled, the SPI bus transfers end after the transmit FIFO is empty and LPSPI is idle.

The HREQ input is only checked when PCS is negated.

33.3.1.1.2 Circular FIFO

The transmit and command FIFO supports a circular FIFO feature. This feature enables the LPSPI controller to (periodically) repeat a short data transfer that fits within the transmit and command FIFO, without requiring additional FIFO accesses. When the circular FIFO is enabled (CFGR0[CIRFIFO] = 1), the current state of the FIFO read pointer is saved and the status flags are not updated. After the FIFO is empty and LPSPI is idle, the FIFO read pointer is restored with the saved version, so the contents of the transmit and command FIFO are not permanently pulled from the FIFO when Circular FIFO mode is enabled.

33.3.1.2 Receive FIFO and data match

The receive FIFO stores received data during SPI bus transfers. When TCR[RXMSK] = 1, the received data is discarded instead of being stored in the receive FIFO:

- Received data is written to the receive FIFO when the last bit of the word is sampled.
- If the transmit FIFO is empty during a multiple-word or continuous transfer, then the receive data is written to the receive FIFO before the transfer stalls (assuming CFGR1[NOSTALL] = 0) while waiting for new transmit data or for a command word to be written.

LPSPI provides a receive data match function that can match received data against one of the two words in DMR0 and DMR1, or against a masked data word. You can also configure the received data match function to compare only the first one or two received data words since the start of the frame:

- Received data that is already discarded because of TCR[RXMSK] cannot cause the data match flag to set, and delays the receive data match on the first received data word, until all discarded data is received.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

- You can configure the receive data match function to discard all received data until a data match is detected, using CFGR0[RDMO].
- After a receive data match, to allow all subsequent data to be received, write 0 to CFGR0[RDMO], and then write 0 to SR[DMF].

33.3.1.3 Timing parameters

The timing parameters that are used for all SPI bus transfers are relative to the LPSPI functional clock divided by the selection specified in TCR[PRESCALE]. Although you can only change Clock Configuration (CCR) when LPSPI is disabled (CR[MEN] = 0), to support interfacing with different peripheral devices at different frequencies, you can change the TCR[PRESCALE] selection between SPI bus transfers by using Transmit Command (TCR).

NOTE

The minimum value shown in Table 197 is the minimum counter value, but the values of Clock Configuration (CCR) must also satisfy the data sheet specs based on the LPSPI functional clock frequency and prescaler value.

Table 197. Timing parameters

Clock Configuration (CCR) Clock Configuration 1 (CCR1)		Description	Minimum value	Maximum value
Field	Name			
SCKSET	SCK setup phase	Configures the SCK setup phase to (SCKSET + 1) cycles. The setup phase is the SCK high period when either CPHA = 0 and CPOL = 1, or CPHA = 1 and CPOL = 0. Otherwise, it is the SCK low period. The SCK period is defined as (SCKSET + SCKHLD + 2) and the duty cycle is the difference between SCKSET and SCKHLD.	0 (1 cycle)	255 (256 cycles)
SCKHLD	SCK hold phase	Configures the SCK hold phase to (SCKHLD + 1) cycles. The hold phase is the SCK low period when either CPHA = 0 and CPOL = 1, or CPHA = 1 and CPOL = 0. Otherwise, it is the SCK high period. The SCK period is defined as (SCKSET + SCKHLD + 2) and the duty cycle is the difference between SCKSET and SCKHLD.	0 (1 cycle)	255 (256 cycles)
PCSPCS	PCS-to-PCS delay	Configures the minimum delay between PCS negation and the next PCS assertion to (PCSPCS + PCSPCS + 2) cycles. When the command word is updated between transfers, there is a minimum of (PCSPCS + 1) cycles between the command word update and any change on PCS pins.	0 (2 cycles)	255 (512 cycles)
SCKSCK	SCK-to-SCK delay	Configures the delay during a continuous transfer between the last SCK edge of a frame and the first SCK edge of the continuing frame to (SCKSCK + 1) cycles. This is useful when the external peripheral requires a large delay between different words of an SPI bus transfer.	0 (1 cycle)	255 (256 cycles)

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table 197. Timing parameters (continued)

Clock Configuration (CCR) Clock Configuration 1 (CCR1)		Description	Minimum value	Maximum value
Field	Name			
PCSSCK	PCS-to-SCK delay	Configures the minimum delay between PCS assertion and the first SCK edge to (PCSSCK + 1) cycles.	0 (1 cycle)	255 (256 cycles)
SCKPCS	SCK-to-PCS delay	Configures the minimum delay between the last SCK edge and the PCS negation to (SCKPCS + 1) cycles.	0 (1 cycle)	255 (256 cycles)

Figure 140 shows the timing settings controlled by:

- TCR[CPHA]
- TCR[CPOL]
- CCR[SCKPCS]
- CCR[PCSSCK]
- CCR1[SCKSET]
- CCR1[SCKHLD]

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

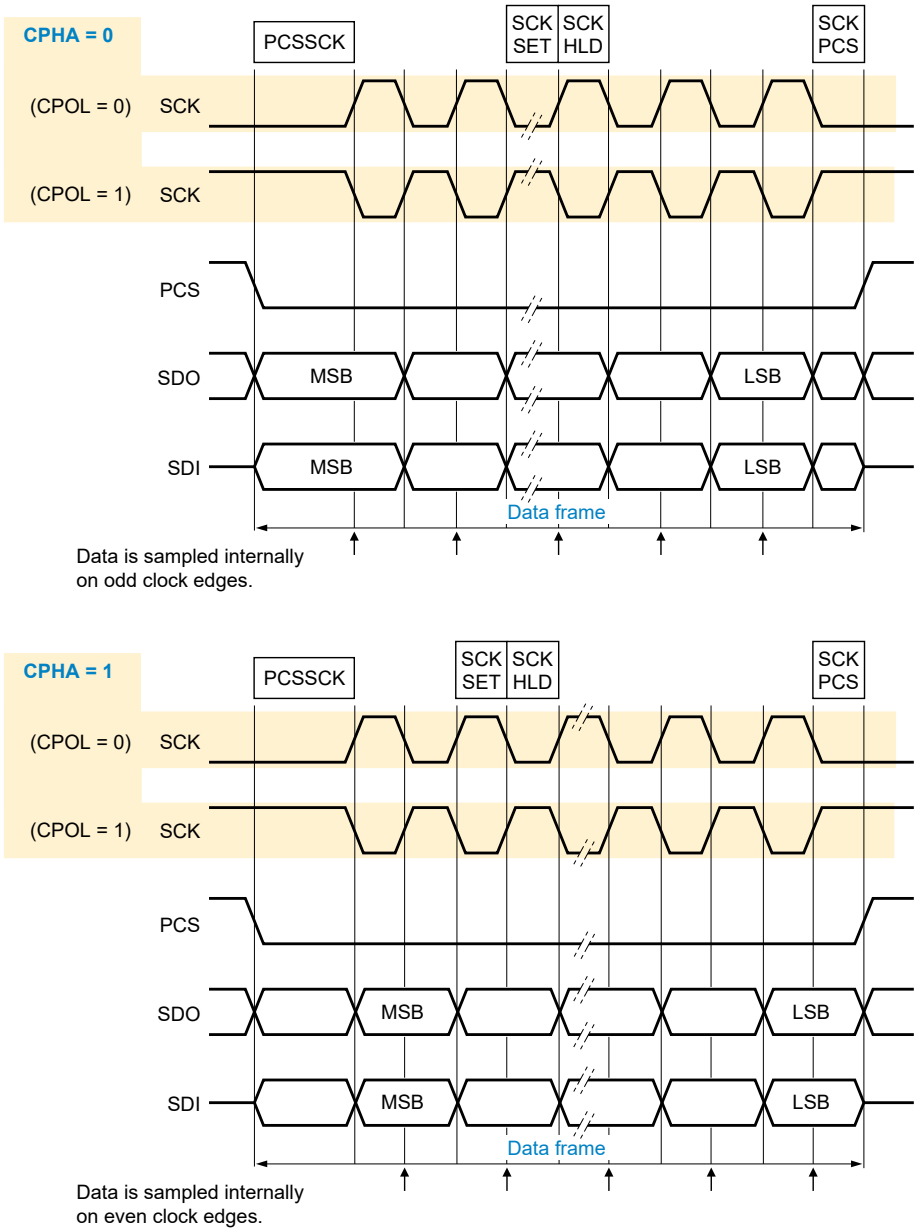


Figure 140. Clock phase (TCR[CPHA]) timing diagram example

To configure for a baud rate of 10 MHz with 50/50 duty cycle and with a functional clock frequency of 100 MHz, use the following settings:

- CCR1[SCKSET] = 0x4 (5 cycles)
- CCR1[SCKHLD] = 0x4 (5 cycles)
- CCR1[PCSPCS] = 0x8 (10 cycles)
- CCR1[SCKSCK] = 0x4 (5 cycles)
- CCR[PCSSCK] = 0x4 (5 cycles)
- CCR[SCKPCS] = 0x4 (5 cycles)
- TCR[PRESCALE] = 0x0 (divide by 1)

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

33.3.1.4 Pin configuration

Following are the pin configuration settings for half-duplex transfers:

- To swap directions or to support half-duplex transfers on the same pin, you can configure the SIN and SOUT pins using CFGR1[PINCFG].
- To specify whether an output data pin (SOUT, for example) 3-states when PCS is negated, or if the output data pin retains the last value, use CFGR1[OUTCFG].
- When configuring half-duplex transfers, you must configure the output data pins to 3-state when PCS is negated (CFGR1[OUTCFG] = 1).
- When performing half-duplex 2-bit transfers, you can write any value to CFGR1[PCSCFG].
- When performing half-duplex 4-bit transfers, you must write 1h to CFGR1[PCSCFG].

33.3.1.5 Clock loopback

Configure the LPSPI controller to use one of the following clocks to sample the input data:

- The SCK output clock
- A delayed version of the SCK output clock

The delayed version of the SCK is chosen by the SCK pin output delay, plus the SCK pin input delay, and is selected by writing 1 to CFGR1[SAMPLE]. Enabling the loopback version of the SCK pin can improve the setup time of the input data from the peripheral.

See the chip data sheet for the specific input setup time in Controller Loopback mode.

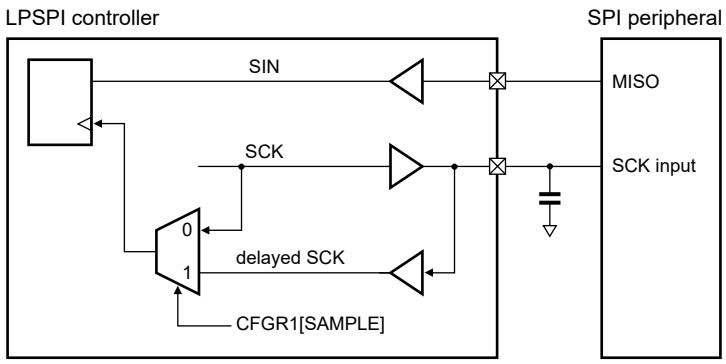


Figure 141. Clock loopback

33.3.2 Peripheral mode

LPSPI Peripheral mode:

- Uses the same shift register and logic that Controller mode uses.
- Does not use Clock Configuration (CCR).
- Requires Transmit Command (TCR) to remain static (unchanged) during SPI bus transfers.

33.3.2.1 Transmit and command FIFO commands

You must initialize Transmit Command (TCR) before enabling LPSPI in Peripheral mode, although this register is not updated until after LPSPI is enabled. After LPSPI is enabled, you must make changes to this register only when LPSPI is idle. In Peripheral mode, the LPSPI command word in this register controls SPI attributes. Before the PCS input asserts, the transmit FIFO must be filled with transmit data, or the transmit error flag sets.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table 198. Command word in Peripheral mode

Transmit Command (TCR)		Description
Field	Name	
CPOL	Clock polarity	Specifies the polarity of the external SCK input.
CPHA	Clock phase	Specifies the clock phase of transfer.
PRESCALE	Prescaler value	Specifies the LPSPI functional clock prescaler.
PCS	Peripheral chip select	Specifies which PCS is used. The polarity of PCS is static and configured by CFGR1[PCSPOL]. If CFGR1[PCSCFG] is not equal to zero, then do not select the PCS[3:2] pins.
LSBF	LSB first	Specifies whether LSB (bit 0) or MSB (bit 31 for a 32-bit word) is transmitted or received first.
BYSW	Byte swap	Enables byte swap on each 32-bit word when transmitting and receiving data. Byte swapping can be useful when interfacing with devices that organize data as big-endian.
CONT	Continuous transfer	When continuous transfer is selected in Peripheral mode, after the number of bits indicated by FRAMESZ are transferred, LPSPI passes through and transmits the received data until the next PCS negation. Whatever is shifted in on the receive data is shifted out as transmit data considering that there is a 32-bit shift register.
CONTC	Continuing command	When the continuing command is enabled in Peripheral mode, after the number of bits indicated by FRAMESZ are transferred, RXMSK is considered equal to 1 and TXMSK is considered equal to 0 until the next PCS negation. CONTC can be used to change the direction of a transfer after the number of bits indicated by FRAMESZ.
RXMSK	Receive data mask	Masks the receive data; LPSPI does not store masked receive data in the receive FIFO or perform receive data matching. This option is useful for half-duplex transfers or to specify which fields are compared during receive data matching.
TXMSK	Transmit data mask	Masks the transmit data so that the masked transmit data is not pulled from transmit FIFO, and the output data pin is 3-stated (unless otherwise specified in CFGR1[OUTCFG]). This option is useful for half-duplex transfers.
WIDTH	Transfer width	Specifies the number of bits shifted on each SCK pulse: • 1-bit transfers support traditional SPI bus transfers in either half-duplex or full-duplex data formats. • 2-bit and 4-bit half-duplex transfers are useful for interfacing with QuadSPI memory devices, and at least either TCR[TXMSK] or TCR[RXMSK] must be 1.
FRAMESZ	Frame size	Specifies the frame size in number of bits equal to (FRAMESZ + 1): • The minimum frame size is 8 bits. • If the frame size is larger than 32 bits, then the frame is divided into multiple words of 32 bits; each word is loaded from the transmit FIFO and stored in the receive FIFO separately. • If the size of the frame is not divisible by 32, then the last load of the transmit FIFO and store of the receive FIFO contain the remainder bits. For example, a 72-bit transfer consists of three words: the first and second words are 32 bits, and the third word is 8 bits.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

33.3.2.2 Receive FIFO and data match

The receive FIFO stores receive data during SPI bus transfers. When TCR[RXMSK] = 1, the received data is discarded instead of storing the received data in the receive FIFO.

Receive data supports a receive data match function that can match received data against one of the two words in DMR0 and DMR1 or against a masked data word. You can also configure the data match function to compare only the first one or two received data words since the start of the frame.

- Received data that is already discarded because TCR[RXMSK] = 1 cannot cause the data match to set, and delays the match on the first received data word, until all discarded data is received.
- By using CFGR0[RDMO], you can also configure the receiver match function to discard all received data until a data match is detected.
- After a receive data match, to allow all subsequent data to be received, first write 0 to CFGR0[RDMO], then clear SR[DMF].

33.3.2.3 Partial received word

When the PCS pin deasserts and the receive shift register shifts in a partial word, you can configure the receive shift register to either discard the partial word or to store it in the receive FIFO. You must specify this using CFGR1[PARTIAL].

A partial word is defined as less than TCR[FRAMESZ] bits (when TCR[FRAMESZ] is equal or less than 32 bits, or it is the last word in a multi-word frame) or less than 32 bits (when TCR[FRAMESZ] is greater than 32 bits and not the last word in a multi-word frame).

A single-bit frame is not supported. A partial received word of 1 bit is supported, but a partial received frame of 1 bit is not supported.

33.3.2.4 Clocked interface

LPSPI supports interfacing with external controllers that provide only clock and data pins (PCS is not required). This interface requires:

- Writing 1 to TCR[CPHA] (data is changed on the leading edge of SCK and captured on the following edge).
- Configuring the PCS input to be always asserted (CFGR1[PCSPOL*n*] = 1). For example, to configure PCS[0] to be always asserted, write 1 to PCSPOL[0], and do not configure PCS[0] in the pin muxing. The chip-level drives PCS to a certain value (ideally 1); you could use CFGR1[PCSPOL*n*] to invert that value.
- Writing 1 to CFGR1[AUTOPCS] to enable automatic PCS generation. When CFGR1[AUTOPCS] = 1, a minimum of four LPSPI functional clock cycles (divided by the selection specified in TCR[PRESCALE]) is required between the last SCK edge of one word and the first SCK edge of the next word.

33.3.3 Debug mode

Table 199. Debug mode

Chip mode	LPSPI operation
Debug (the core is in Debug or Halted mode)	Can continue operating in Debug mode, if CR[DBGEN] = 1

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

33.3.4 Clocking

Table 200. LPSPI clocks

Type of clock	Description
Functional	• Asynchronous to the bus clock. • If the LPSPI functional clock remains enabled in low-power modes, then LPSPI can perform SPI bus transfers and low-power wakeups in both Controller and Peripheral modes. • LPSPI divides the functional clock by a prescaler; the resulting frequency must be at least two times faster than the SPI external clock frequency (SCK).
External	• The LPSPI shift register is clocked directly by the SCK clock. • How the SCK clock is generated or supplied depends on the mode (Controller or Peripheral): — In Controller mode, the SCK clock is generated internally. — In Peripheral mode, the SCK clock is supplied externally.
Bus	The bus clock is only used for bus accesses to the LPSPI control and configuration registers. The bus clock frequency must be high enough to support the data bandwidth requirements of the LPSPI registers, including the FIFOs.

See the chip-specific LPSPI information for more.

33.3.5 Reset

Table 201. LPSPI resets

Type of reset	Description
Chip	Resets the LPSPI logic and registers to their default states.
Software	• Resets the LPSPI logic and registers to their default states, except for the Control register. • The LPSPI software reset is controlled using CR[RST].
FIFO	• Resets the transmit and command FIFO and the receive FIFO. • CR[RTF] and CR[RRF] are write-only. • After being reset, FIFO is empty.

33.3.6 Interrupts and DMA requests

The following table lists sources (status flags) that can generate LPSPI interrupts and LPSPI transmit and receive DMA requests.

Table 202. Interrupts and DMA requests

Status (SR)		Description	Can generate		
Status flag	Name		Interrupt?	DMA request?	Low-power wake-up?
TDF	Transmit data flag	Indicates that data can be written to transmit FIFO, as configured by the transmit FIFO watermark, FCR[TXWATER].	Y	TX	Y

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table 202. Interrupts and DMA requests (continued)

Status (SR)		Description	Can generate		
Status flag	Name		Interrupt?	DMA request?	Low-power wake-up?
RDF	Receive data flag	Indicates that data can be read from the receive FIFO, as configured by the receive FIFO watermark, FCR[RXWATER].	Y	RX	Y
WCF	Word complete flag	Indicates that the word is complete and the last bit of the word has been sampled.	Y	N	Y
FCF	Frame complete flag	Indicates that the frame is complete and PCS is deasserted.	Y	RX	Y
TCF	Transfer complete flag	Indicates that transfer is complete, PCS is deasserted, and the transmit and command FIFO is empty.	Y	N	Y
TEF	Transmit error flag	Indicates a transmit and command FIFO underrun. In Controller mode, when CFGR1[NOSTALL] = 0 (transfers stall when transmit FIFO is empty), TEF cannot be set.	Y	N	Y
REF	Receive error flag	Indicates a receive FIFO overflow. In Controller mode, when CFGR1[NOSTALL] = 0 (transfers stall when receive FIFO is full), REF cannot be set.	Y	N	Y
DMF	Data match flag	Indicates that the received data matches the configured data match value.	Y	N	Y
MBF	Module busy flag	Indicates that LPSPI is busy performing an SPI bus transfer.	N	N	N

33.3.6.1 End-of-packet DMA transfer

The end-of-packet functionality is designed for serial interfaces where you may not be aware of the size of the transfer in advance and the data is pushed by an external device. Examples include UART receive, I2C Peripheral mode, and SPI Peripheral mode. The end-of-packet processing is intended to ensure that data is not stranded in either the receive FIFO or the DMA receive buffer. Support for end-of-packet processing must be implemented in both the serial interfaces and the DMA controller.

The condition that signals the end of packet is different for each serial interface but the serial peripheral and DMA process it in the same way. For example, UART end of packet is signaled by an idle line condition, I2C end of packet by a stop and/or repeated start condition, and SPI end of packet by PCS negation.

If you configure the serial peripheral to signal the end-of-packet condition to the DMA and the serial interface detects an end-of-packet condition, it asserts the DMA request for the receive FIFO irrespective of the watermark configuration. For larger watermark configurations, this ensures that the last few words of the transfer are first flushed from the receive FIFO.

The DMA then reads the contents of the receive FIFO, depending on the first word in the FIFO:

- If the receive FIFO is empty, the serial interface signals an end-of-packet condition to the DMA controller.
- If the receive FIFO is not empty, but the first word in the FIFO is the start of a new packet, then data is not pulled from the receive FIFO and the serial interface signals an end-of-packet condition to the DMA controller.
- If the receive FIFO is not empty, and the first word in the FIFO is not the start of a new packet, the DMA transfers the receive data as normal.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Because the DMA may be transferring multiple words on each request, the end-of-packet condition persists until the DMA minor loop is complete and no additional data is pulled from the receive FIFO. The status flag that triggered the end-of-packet condition is cleared when the minor loop completes following end of packet being signaled to the DMA controller.

When the DMA detects the end-of-packet condition, it writes all received words up to the end of the packet into the system memory and saves the destination address for the word after the last valid data. The DMA then terminates the channel as if the major loop is complete, including final offsets and optional interrupts, channel linking and scatter/gather. The final destination address can optionally be saved in the system memory or is available in the destination address register.

After the DMA terminates the major loop, no servicing of the receive FIFO occurs until either software or hardware reconfigures the DMA (for example, channel linking or scatter/gather). This delay must be minimized to avoid receiver FIFO overrun. NXP does not recommend automatic DMA end-of-packet processing when there are only a few words transferred between end-of-packet conditions. That is because the DMA spends more time processing the end of packet than transferring the data. For example, the UART idle line length must be increased as needed to avoid an excessive number of idle conditions.

33.3.6.2 DMA support registers

To support efficient DMA transfers to the transmit and control FIFO, an alias register supports 32-bit write access to the Transmit Command (TCR) and an alias region supports incrementing 32-bit write accesses to the Transmit Data (TDR).

- Transmit Command Burst (TCBR) is a 32-bit alias register for writing to TCR.
- Transmit Data Burst (TDBR0 - TDBR127) is a 512-byte alias region that supports writing to the TDR.

The burst alias locations are contiguous. A DMA transfer can start by writing to the TCBR register to initialize the transfer and then increment into the TDBR region without changing the DMA transfer size. The alias registers can also be used by the DMA to perform burst transfers when accessing the transmit FIFO.

The transmit FIFO prevents writes that overflow the FIFO, but this prevention does not signal an error. Do not perform writes to the TDBR unless there is sufficient room in the transmit FIFO.

Receive Data Burst (RDBR0 - RDBR127) is a 512-byte alias region that supports reading the Receive Data. This can be used by the DMA to perform burst transfers when accessing the receive FIFO.

33.3.7 Peripheral triggers

The connection of the LPSPI peripheral triggers with other peripherals depends on the device that is used.

Table 203. Peripheral triggers

Type of trigger	Description	Additional information
Frame output	The frame output trigger: • Asserts at the end of each frame (when PCS deasserts). • Remains asserted for one cycle of the LPSPI functional clock divided by the configuration defined in TCR[PRESCALE].	LPSPI generates two output triggers that can be connected to other peripherals on the chip.
Word output	The word output trigger: • Asserts at the end of each received word. • Remains asserted for one cycle of the LPSPI functional clock divided by the configuration defined in TCR[PRESCALE].	
Input	To control the start of an LPSPI bus transfer, the LPSPI input trigger can be selected instead of the HREQ input:	

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table 203. Peripheral triggers (continued)

Type of trigger	Description	Additional information
	• The LPSPI input trigger is synchronized, and must assert for at least two cycles of the LPSPI functional clock divided by the configuration defined in TCR[PRESCALE] so that the input trigger can be detected. • When LPSPI is busy, the HREQ input (and therefore the LPSPI input trigger) is ignored. • When LPSPI is busy, both the HREQ and LPSPI input triggers are ignored. They are used to start a new transfer when LPSPI is idle.	

33.4 External signals

Table 204. External signals

Signal	Name	Description	I/O
SCK	Serial clock	• Input in Peripheral mode • Output in Controller mode	I/O
PCS[0]	Peripheral chip select	• Input in Peripheral mode • Output in Controller mode	I/O
PCS[1]/HREQ	Peripheral chip select or host request	Host request pin is selected when CFGR0[HREN] = 1 and CFGR0[HRSEL] = 0: • Input in either Peripheral mode or when used as controller host request • Output in either Controller mode or when used as peripheral host request	I/O
PCS[2]/DATA[2]	Peripheral chip select or data pin 2 during parallel data transfers	When CFGR1[PCSCFG] = 0: • Input in Peripheral mode • Output in Controller mode When CFGR1[PCSCFG] = 1: • Input in half-duplex parallel data receive transfers • Output in half-duplex parallel data transmit transfers	I/O
PCS[3]/DATA[3]	Peripheral chip select or data pin 3 during parallel data transfers	When CFGR1[PCSCFG] = 0: • Input in Peripheral mode • Output in Controller mode When CFGR1[PCSCFG] = 1: • Input in half-duplex parallel data receive transfers	I/O

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table 204. External signals (continued)

Signal	Name	Description	I/O
		• Output in half-duplex parallel data transmit transfers	
SOUT/DATA[0]	Serial data output	Can be configured as serial data input signal (used as data pin 0 in half-duplex parallel data transfers)	I/O
SIN/DATA[1]	Serial data input	Can be configured as serial data output signal (used as data pin 1 in half-duplex parallel data transfers)	I/O

33.5 Initialization

This module does not require initialization.

33.6 Memory map and registers

NOTE

- Writing to a read-only register or reading a write-only register can cause bus errors.
- LPSPI does not check values programmed in registers for validity, so you must take care to write valid values only.

33.6.1 LPSPI register descriptions

33.6.1.1 LPSPI memory map

LPSPI0 base address: 4009_C000h

Offset	Register	Width (In bits)	Access	Reset value
0h	Version ID (VERID)	32	R	0200_0004h
4h	Parameter (PARAM)	32	R	0004_0202h
10h	Control (CR)	32	RW	0000_0000h
14h	Status (SR)	32	RW	0000_0001h
18h	Interrupt Enable (IER)	32	RW	0000_0000h
1Ch	DMA Enable (DER)	32	RW	0000_0000h
20h	Configuration 0 (CFGR0)	32	RW	0000_0000h
24h	Configuration 1 (CFGR1)	32	RW	0000_0000h
30h	Data Match 0 (DMR0)	32	RW	0000_0000h
34h	Data Match 1 (DMR1)	32	RW	0000_0000h
40h	Clock Configuration (CCR)	32	RW	0000_0000h
44h	Clock Configuration 1 (CCR1)	32	RW	0000_0000h
58h	FIFO Control (FCR)	32	RW	0000_0000h

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Offset	Register	Width (In bits)	Access	Reset value
5Ch	FIFO Status (FSR)	32	R	0000_0000h
60h	Transmit Command (TCR)	32	RW	0000_001Fh
64h	Transmit Data (TDR)	32	W	0000_0000h
70h	Receive Status (RSR)	32	R	0000_0002h
74h	Receive Data (RDR)	32	R	0000_0000h
78h	Receive Data Read Only (RDROR)	32	R	0000_0000h
3FCh	Transmit Command Burst (TCBR)	32	W	0000_0000h
400h - 5FCh	Transmit Data Burst (TDBR0 - TDBR127)	32	W	0000_0000h
600h - 7FCh	Receive Data Burst (RDBR0 - RDBR127)	32	R	0000_0000h

33.6.1.2 Version ID (VERID)

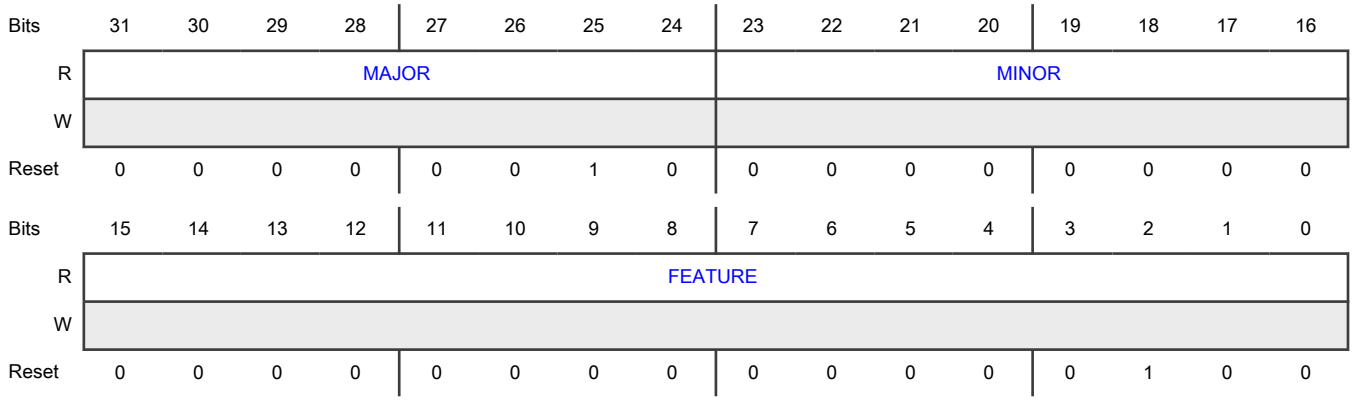
Offset

Register	Offset
VERID	0h

Function

Contains version numbers for the module design and feature set.

Diagram



All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Fields

Field	Function
31-24 MAJOR	Major Version Number Indicates the major version number of the module specification.
23-16 MINOR	Minor Version Number Indicates the minor version number of the module specification.
15-0 FEATURE	Module Identification Number Indicates the feature set number 0000_0000_0000_0100b - Standard feature set supporting a 32-bit shift register. All other values are reserved.

33.6.1.3 Parameter (PARAM)

Offset

Register	Offset
PARAM	4h

Function

Contains:

- Number of PCS pins.
- Receive FIFO size.
- Transmit FIFO size.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0								PCSNUM							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RXFIFO								TXFIFO							
W																
Reset	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Fields

Field	Function
31-24 —	Reserved
23-16 PCSNUM	PCS Number Indicates the number of PCS pins supported.
15-8 RXFIFO	Receive FIFO Size Indicates the maximum number of words in the receive FIFO. The maximum number of words is 2^{RXFIFO}.
7-0 TXFIFO	Transmit FIFO Size Indicates the maximum number of words in the transmit FIFO. The maximum number of words is 2^{TXFIFO}.

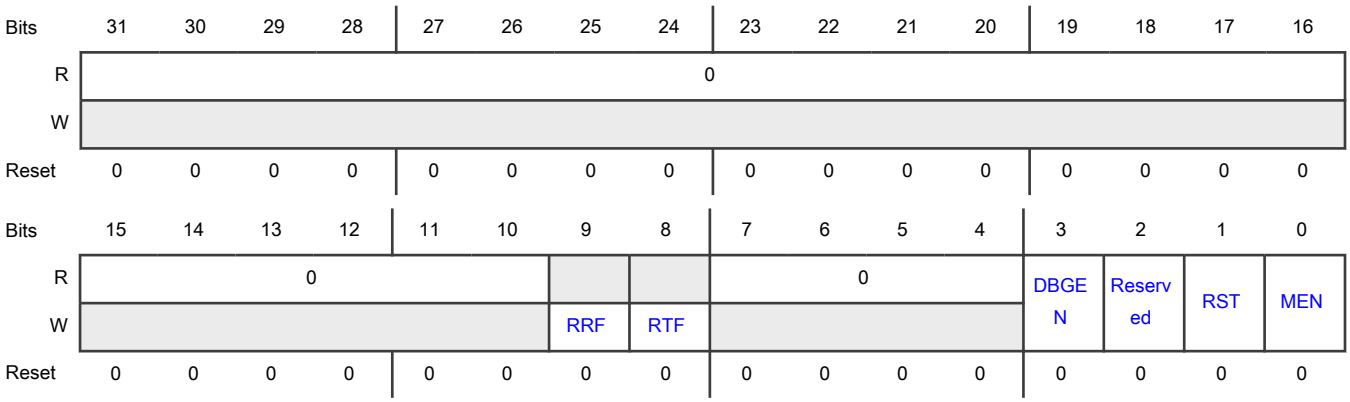
33.6.1.4 Control (CR)

Offset

Register	Offset
CR	10h

Function
Contains fields that control the module operation.

Diagram



Fields

Field	Function
31-10	Reserved

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
—	
9 RRF	Reset Receive FIFO Deletes all entries in the receive FIFO. This field always reads 0. 0b - No effect 1b - Reset
8 RTF	Reset Transmit FIFO Deletes all entries in the transmit FIFO. This field always reads 0. 0b - No effect 1b - Reset
7-4 —	Reserved
3 DBGEN	Debug Enable Enables LPSPI when the CPU is in Debug mode. If this field is 0, LPSPI is disabled when the CPU is halted; the PCS pin is deasserted after the transmit FIFO is empty regardless of the state of Transmit Command (TCR). You must update this field only when LPSPI is disabled (MEN = 0). 0b - Disable 1b - Enable
2 —	Reserved
1 RST	Software Reset Resets all internal logic and registers, except Control (CR). The reset takes effect immediately and remains asserted until you write 0 to it. There is no minimum delay required before clearing the software reset by writing 0. 0b - Not reset 1b - Reset
0 MEN	Module Enable Enables the module. After writing 0, MEN remains set until LPSPI has completed the current transfer and is idle. 0b - Disable 1b - Enable

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

33.6.1.5 Status (SR)

Offset

Register	Offset
SR	14h

Function

Contains data flow status.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0							MBF	0							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		DMF	REF	TEF	TCF	FCF	WCF	0						RDF	TDF
W			W1C	W1C	W1C	W1C	W1C	W1C								
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Fields

Field	Function
31-25 —	Reserved
24 MBF	Module Busy Flag Indicates, in Controller mode, whether there is data to transmit and LPSPI is able to transmit (for example, the HREQ pin is asserted). The HREQ pin deasserts after the PCS pin deasserts and the LPSPI controller has waited for half the time specified in CCR[DBT] with no new data to transmit. Peripheral mode sets this flag when LPSPI is enabled and PCS is asserted. 0b - LPSPI is idle 1b - LPSPI is busy
23-14 —	Reserved
13 DMF	Data Match Flag Indicates whether the received data matches DMR0[MATCH0] and/or DMR1[MATCH1] (as configured by CFGR1[MATCFG]).

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	NOTE This field behaves differently for register reads and writes. When reading 0b - No match 1b - Match When writing 0b - No effect 1b - Clear the flag
12 REF	Receive Error Flag Indicates a receive FIFO overflow error. When this flag is set: 1. End the transfer. 2. Empty the receive FIFO. 3. Clear this flag. 4. Restart the transfer from the beginning. **NOTE** This field behaves differently for register reads and writes. When reading 0b - No overflow 1b - Overflow When writing 0b - No effect 1b - Clear the flag
11 TEF	Transmit Error Flag Indicates a transmit FIFO underrun error. When this flag is set: 1. End the transfer. 2. Clear this flag. 3. Restart the transfer from the beginning. **NOTE** This field behaves differently for register reads and writes. When reading 0b - No underrun 1b - Underrun

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	When writing 0b - No effect 1b - Clear the flag
10 TCF	Transfer Complete Flag Indicates, in Controller mode, whether all transfers are complete and LPSPI has returned to the Idle state and the transmit FIFO is empty. **NOTE** This field behaves differently for register reads and writes. When reading 0b - Not complete 1b - Complete When writing 0b - No effect 1b - Clear the flag
9 FCF	Frame Complete Flag Indicates whether a frame transfer is complete after PCS deasserts. **NOTE** This field behaves differently for register reads and writes. When reading 0b - Not complete 1b - Complete When writing 0b - No effect 1b - Clear the flag
8 WCF	Word Complete Flag Indicates whether the last bit of a received word is sampled. **NOTE** This field behaves differently for register reads and writes. When reading 0b - Not complete 1b - Complete When writing

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	0b - No effect 1b - Clear the flag
7-2 —	Reserved
1 RDF	Receive Data Flag Indicates whether the number of words in the receive FIFO is greater than the value in FCR[RXWATER]. 0b - Receive data not ready 1b - Receive data ready
0 TDF	Transmit Data Flag Indicates whether the number of words in the transmit FIFO is equal to or less than the value in FCR[TXWATER]. 0b - Transmit data not requested 1b - Transmit data requested

33.6.1.6 Interrupt Enable (IER)

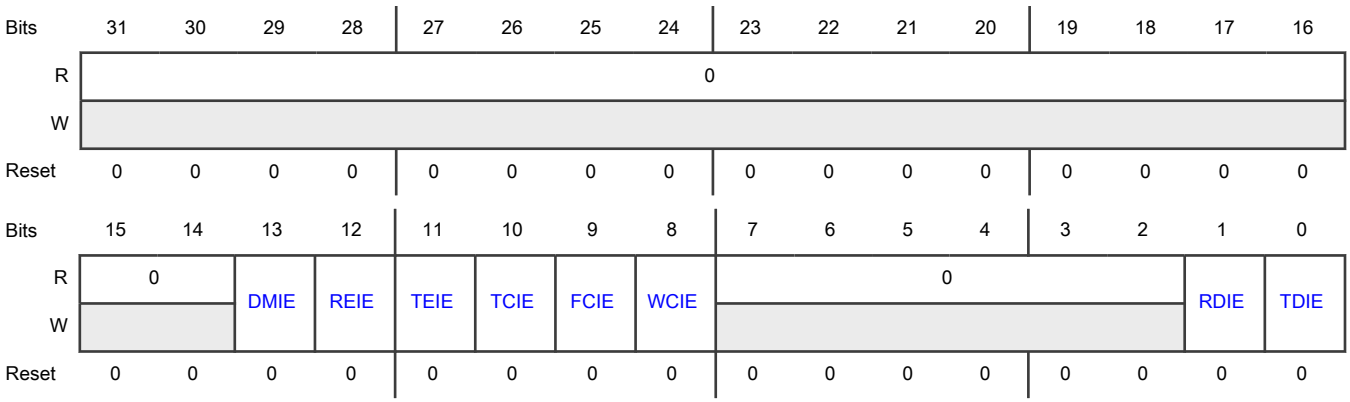
Offset

Register	Offset
IER	18h

Function

Enables interrupts based on data flow and errors.

Diagram



All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Fields

Field	Function
31-14 —	Reserved
13 DMIE	Data Match Interrupt Enable Enables the data match interrupt. 0b - Disable 1b - Enable
12 REIE	Receive Error Interrupt Enable Enables the receive error interrupt. 0b - Disable 1b - Enable
11 TEIE	Transmit Error Interrupt Enable Enables the transmit error interrupt. 0b - Disable 1b - Enable
10 TCIE	Transfer Complete Interrupt Enable Enables the transfer complete interrupt. 0b - Disable 1b - Enable
9 FCIE	Frame Complete Interrupt Enable Enables the frame complete interrupt. 0b - Disable 1b - Enable
8 WCIE	Word Complete Interrupt Enable Enables the word complete interrupt. 0b - Disable 1b - Enable
7-2 —	Reserved
1 RDIE	Receive Data Interrupt Enable Enables the receive data interrupt.

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	0b - Disable 1b - Enable
0 TDIE	Transmit Data Interrupt Enable Enables the transmit data interrupt. 0b - Disable 1b - Enable

33.6.1.7 DMA Enable (DER)

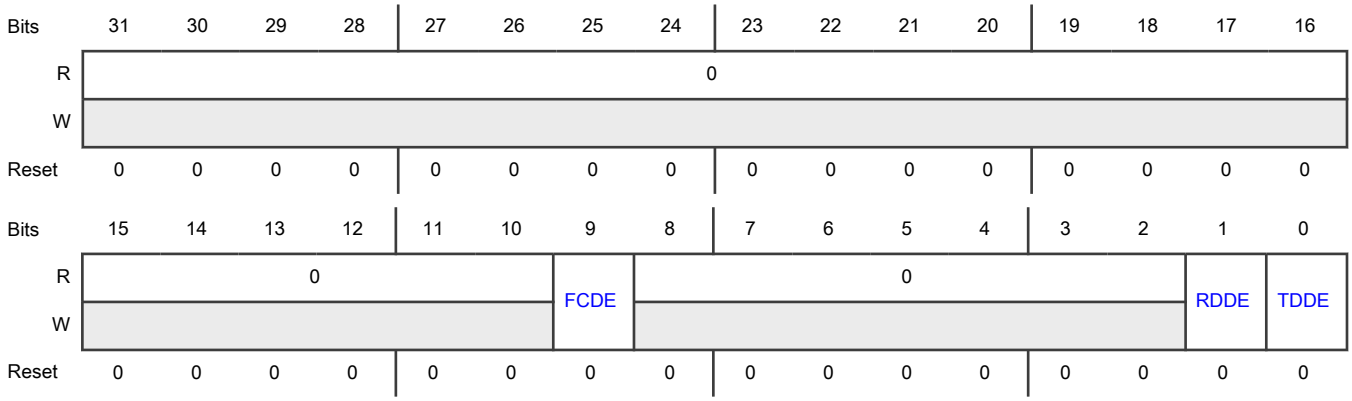
Offset

Register	Offset
DER	1Ch

Function

Enables the DMA data flow.

Diagram



Fields

Field	Function
31-10 —	Reserved
9 FCDE	Frame Complete DMA Enable

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	Enables DMA end-of-packet processing. After the last word of a frame is read from the receive data FIFO, reading the receive data FIFO returns an end-of-packet signal with the receive data forced to FFFF_FFFFh. This continues until the DMA minor loop completes, and then SR[FCF] deasserts if the receive FIFO is empty or if LPSPI is busy (SR[MBF] = 1). 0b - Disable 1b - Enable
8-2 —	Reserved
1 RDDE	Receive Data DMA Enable Enables the receive data DMA. 0b - Disable 1b - Enable
0 TDDE	Transmit Data DMA Enable Enables the transmit data DMA. 0b - Disable 1b - Enable

33.6.1.8 Configuration 0 (CFGR0)

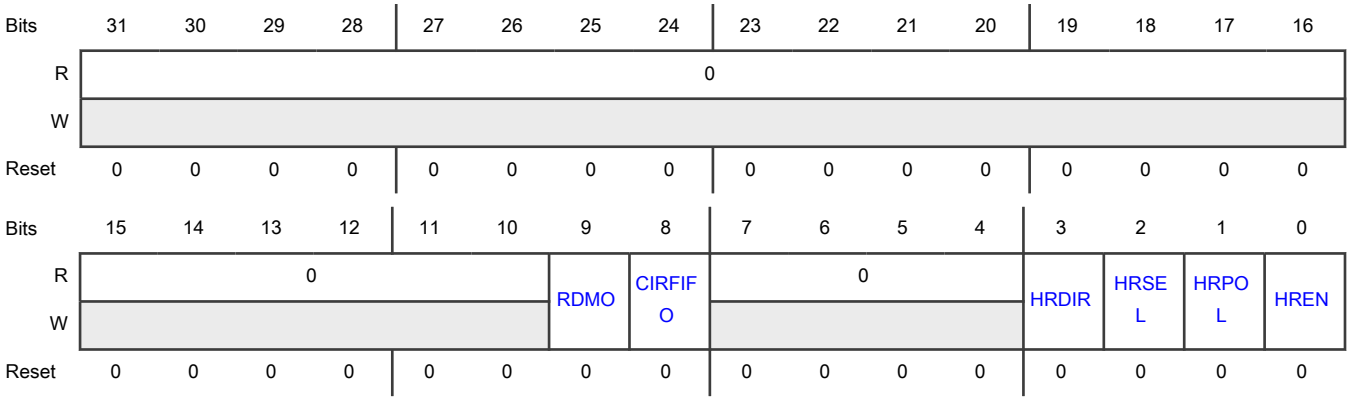
Offset

Register	Offset
CFGR0	20h

Function

Includes fields to configure LPSPI.

Diagram



All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Fields

Field	Function
31-10 —	Reserved
9 RDMO	Receive Data Match Only Enables receive data match. When enabled, all received data that does not cause SR[DMF] to assert is discarded: • Write 1 to this field when LPSPI is idle and SR[DMF] = 0. • After SR[DMF] = 1, this field is ignored. • To ensure that no receive data is lost when disabling RDMO, write 0 to this field before clearing SR[DMF]. See CFGR1[MATCFG] for the received data matching options. When disabled, all received data is stored in the receive FIFO. 0b - Disable 1b - Enable
8 CIRFIFO	Circular FIFO Enable Enables circular FIFO. When enabled, the transmit FIFO read pointer is saved to a temporary register. The transmit FIFO is emptied as in normal operation, but when LPSPI is idle and the transmit FIFO is empty, the read pointer value is restored from the temporary register. This restoring of the read pointer causes the contents of the transmit FIFO to be cycled through repeatedly. **NOTE** The read pointer is restored for as long as this field is 1. Writing additional words to the FIFO when this field is 1 adds them to the end of the FIFO, up to the size of the transmit FIFO. 0b - Disable 1b - Enable
7-4 —	Reserved
3 HRDIR	Host Request Direction Specifies the direction of the HREQ pin. You must configure the HREQ pin only as an output when LPSPI is in Peripheral mode. The HREQ pin direction must be an input for Controller mode. 0b - Input 1b - Output
2 HRSEL	Host Request Select Specifies the source of the host request input. When the host request function is enabled with the HREQ pin, the PCS[1] function is disabled.

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	0b - HREQ pin 1b - Input trigger
1 HRPOL	Host Request Polarity Specifies the polarity of the HREQ pin or input trigger. 0b - Active high 1b - Active low
0 HREN	Host Request Enable Enables LPSPI, in Controller mode, to start a new SPI bus transfer only if the host request input is asserted. When LPSPI is busy, the host request input is ignored. In Peripheral mode, causes the HREQ output pin to assert when data is available to be transmitted. 0b - Disable 1b - Enable

33.6.1.9 Configuration 1 (CFGR1)

Offset

Register	Offset
CFGR1	24h

Function

Includes fields to configure LPSPI. You must write to this register only when LPSPI is disabled.

In addition to pin and output configurations, this register contains match configuration details; the following table shows match conditions specified in MATCFG.

Table 205. Match conditions for CFGR1[MATCFG]

Condition	Description
Match first data word with compare word	Match if first data word equals MATCH0 logically ORed with MATCH1 first_data_word == (MATCH0 || MATCH1)
Match any data word with compare word	Match if any data word equals MATCH0 logically ORed with MATCH1 any_data_word == (MATCH0 || MATCH1)
Sequential match, first data word	Match if first data word equals MATCH0, and second data word equals MATCH1 (first_data_word == MATCH0) && (second_data_word == MATCH1)
Sequential match, any data word	Match if any data word equals MATCH0, and the next data word equals MATCH1

Table continues on the next page...

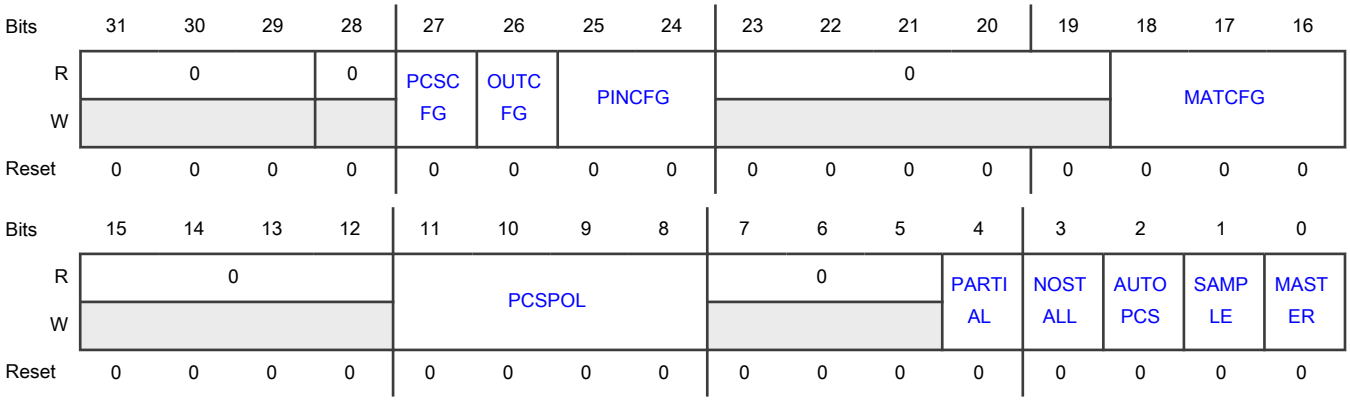
All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table 205. Match conditions for CFGR1[MATCFG] (continued)

Condition	Description
	(any_data_word == MATCH0) && (next_data_word == MATCH1)
Match first data word (masked) with compare word (masked)	Match if first data word logically ANDed with MATCH1 equals MATCH0 logically ANDed with MATCH1 (first_data_word && MATCH1) == (MATCH0 && MATCH1)
Match any data word (masked) with compare word (masked)	Match if any data word logically ANDed with MATCH1 equals MATCH0 logically ANDed with MATCH1 (any_data_word && MATCH1) == (MATCH0 && MATCH1)

Diagram



Fields

Field	Function
31-29 —	Reserved
28 —	Reserved
27 PCSCFG	Peripheral Chip Select Configuration Specifies PCS pin configuration. When performing parallel transfers, you must configure this field to enable the desired transfer. 0b - PCS[3:2] configured for chip select function 1b - PCS[3:2] configured for half-duplex 4-bit transfers (PCS[3:2] = DATA[3:2])
26 OUTCFG	Output Configuration Specifies whether the output data is 3-stated between accesses (when PCS is deasserted). When performing half-duplex transfers, this field must be 1.

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	0b - Retain last value 1b - 3-stated
25-24 PINCFG	Pin Configuration Specifies the pins used for input and output data during serial transfers. **NOTE** When performing parallel transfers, this field must be 00b or 10b. 00b - SIN is used for input data; SOUT is used for output data 01b - SIN is used for both input and output data; only half-duplex serial transfers are supported 10b - SOUT is used for both input and output data; only half-duplex serial transfers are supported 11b - SOUT is used for input data; SIN is used for output data
23-19 —	Reserved
18-16 MATCFG	Match Configuration Specifies the condition that causes SR[DMF] to assert. See the match conditions listed in Table 1 for more information. **NOTE** When writing to this field, either the old value or new value must be in the disabled state (0). You cannot transition from a nonzero value to another nonzero value. 000b - Match is disabled 001b - Reserved 010b - Match first data word with compare word 011b - Match any data word with compare word 100b - Sequential match, first data word 101b - Sequential match, any data word 110b - Match first data word (masked) with compare word (masked) 111b - Match any data word (masked) with compare word (masked)
15-12 —	Reserved
11-8 PCSPOL	Peripheral Chip Select Polarity Specifies the polarity of each PCS pin. Bit *n* in this field (the least-significant bit is bit 0) corresponds to PCS[*n*]. 0000b - Active low

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	0001b - Active high
7-5 —	Reserved
4 PARTIAL	Partial Enable Specifies whether LPSPI, when in Peripheral mode, stores a partial received word in the receive FIFO, or discards it, when PCS deasserts. See Partial received word for more information. 0b - Discard 1b - Store
3 NOSTALL	No Stall Disables a normal operating feature that causes LPSPI, when in Controller mode, to stall transfers when the transmit FIFO is empty or when the receive FIFO is full. This feature prevents transmit FIFO underruns and receive FIFO overruns. Writing 1 to this field disables this functionality. 0b - Disable 1b - Enable
2 AUTOPCS	Automatic PCS Enables automatic PCS generation. For correct operation in Peripheral mode, LPSPI requires the PCS signal to deassert between frames. Writing 1 to this field generates an internal PCS signal at the end of each transfer word when TCR[CPHA] = 1. When this field is 1, SCK must remain idle for at least four LPSPI functional clock cycles, divided by the prescaler (see TCR[PRESCALE]) selected between each word to ensure correct operation. This field is ignored in Controller mode. **NOTE** Set TCR[CPHA] = 1 before enabling the module by setting CR[MEN] = 1. 0b - Disable 1b - Enable
1 SAMPLE	Sample Point Specifies the SCK clock edge on which LPSPI, when in Controller mode, samples input data. Writing 1 to this field causes LPSPI to sample input data on a delayed loopback SCK clock edge, which improves the setup time when sampling data (see Clock loopback). In this configuration, the input data setup time in Controller mode is equal to the input data setup time in Peripheral mode. In Peripheral mode, this field is ignored. **NOTE** When SAMPLE = 1, both the input and output buffers must be enabled for the SCK pin. Buffers are configured at the chip level.

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	0b - SCK edge 1b - Delayed SCK edge
0 MASTER	Controller Mode Specifies the LPSPI operating mode, Controller or Peripheral. This field directly controls the direction of the SCK and PCS pins. 0b - Peripheral mode 1b - Controller mode

33.6.1.10 Data Match 0 (DMR0)

Offset

Register	Offset
DMR0	30h

Function

Specifies the match data to be used when data matching is enabled. See CFGR1[MATCFG] for the received data matching options.

NOTE

Do not change the value in this register when CFGR1[MATCFG] > 0.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	MATCH0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MATCH0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-0 MATCH0	Match 0 Value Specifies the MATCH0 value to be compared against received data.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

33.6.1.11 Data Match 1 (DMR1)

Offset

Register	Offset
DMR1	34h

Function

Specifies the match data to be used when data matching is enabled. See CFGR1[MATCFG] for the received data matching options.

NOTE

Do not change the value in this register while CFGR1[MATCFG] > 0.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	MATCH1															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MATCH1															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-0 MATCH1	Match 1 Value Specifies the MATCH1 value to be compared against received data.

33.6.1.12 Clock Configuration (CCR)

Offset

Register	Offset
CCR	40h

Function

Contains clock configuration fields that are used only in Controller mode; you can only change them when LPSPI is disabled (CR[MEN] = 0).

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Warning

Writing a 32-bit value to this register overwrites Clock Configuration 1 (CCR1); DBT and SCKDIV always read 0.

To avoid overwriting CCR1, do one of the following:

- Write to all four fields in Clock Configuration (CCR) simultaneously and only once in a 32-bit data.
- Modify the values of CCR[SCKPCS] and/or CCR[PCSSCK]; write only these two upper bytes in a 16-bit data or one of them in an 8-bit data.
- Modify CCR1[PCSPCS] and CCR1[SCKSCK] only or CCR1[SCKSET] and CCR1[SCKHLD] only, write respectively to CCR[DBT] or CCR[SCKDIV] in 8-bit data.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SCKPCS								PCSSCK							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								0							
W	DBT								SCKDIV							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-24 SCKPCS	SCK-to-PCS Delay Configures SCK-to-PCS delay. In Controller mode, this field helps you configure the delay from the last SCK edge to PCS negation: • The delay is equal to (SCKPCS + 1) cycles of the LPSPI functional clock divided by the selected prescaler (see TCR[PRESCALE]). • The minimum delay is one cycle. See Figure 140 for more information.
23-16 PCSSCK	PCS-to-SCK Delay Configures PCS-to-SCK delay. In Controller mode, this field helps you configure the delay from PCS assertion to the first SCK edge: • The delay is equal to (PCSSCK + 1) cycles of the LPSPI functional clock divided by the selected prescaler (see TCR[PRESCALE]). • The minimum delay is one cycle. See Figure 140 for more information.
15-8 DBT	Delay Between Transfers Configures the delay between transfers. Writing to this field updates the contents of CCR1[PCSPCS] and CCR1[SCKSCK].

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
7-0 SCKDIV	SCK Divider Updates the contents of CCR1[SCKSET] and CCR1[SCKHLD]. Baud rate = function clock ÷ (2^PRESCALE × (SCKSET + SCKHLD + 2))

33.6.1.13 Clock Configuration 1 (CCR1)

Offset

Register	Offset
CCR1	44h

Function

Contains clock configuration fields, which are used only in Controller mode. You can change them only when LPSPI is disabled (CR[MEN] = 0).

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SCKSCK								PCSPCS							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SCKHLD								SCKSET							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-24 SCKSCK	SCK Inter-Frame Delay Configures SCK inter-frame delay in Controller mode: • This field helps you configure the delay from the last SCK pulse of a frame and the first SCK pulse of the following frame, in a continuous transfer. • The delay is equal to (SCKSCK + 1) cycles of the LPSPI functional clock divided by the selected prescaler (see TCR[PRESCALE]). • The minimum delay is one cycle.

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	NOTE For backward compatibility, writing to CCR[DBT] updates this field with the value written.
23-16 PCSPCS	PCS to PCS Delay Configures PCS to PCS delay in Controller mode: • This field helps you configure the delay from the PCS negation to the next PCS assertion. • The delay is equal to (PCSPCS + PCSPCS + 2) cycles of the LPSPI functional clock divided by the selected prescaler (see TCR[PRESCALE]). • The minimum delay is two cycles. • Half of the delay (PCSPCS + 1) occurs before PCS assertion and the other half of the delay (PCSPCS + 1) occurs after PCS negation. If the command word is updated between two transfers, then the command word is updated halfway between the PCS negation of the last transfer and PCS assertion of the next transfer. • The command word specifies which PCS signal is used, the polarity and phase of the SCK signal, and the selected prescaler. **NOTE** For backward compatibility, writing to CCR[DBT] updates this field with (DBT÷2) rounded up.
15-8 SCKHLD	SCK Hold Configures the hold phase of the SCK pin in Controller mode: • The hold phase is the delay between the SCK edge that samples the receive data and the SCK edge that drives the transmit data. • It is the SCK low period when CPHA = 0 and CPOL = 1, or CPHA = 1 and CPOL = 0. It is the SCK high period when CPHA = 0, CPOL = 0 and CPHA = 1, CPOL = 1. • The SCK hold phase delay is equal to (SCKHLD + 1) cycles of the LPSPI functional clock divided by the selected prescaler (see TCR[PRESCALE]). • The minimum delay is one cycle. • The SCK period is equal to (SCKSET + SCKHLD + 2) cycles of the LPSPI functional clock divided by the selected prescaler (see TCR[PRESCALE]). • The SCK duty cycle is based on the difference between SCKSET and SCKHLD. You must configure both these fields to the same value for a 50/50 duty cycle. See Figure 140 for more information. **NOTE** For backward compatibility, writing to CCR[SCKDIV] updates this field with (SCKDIV ÷ 2) rounded down.
7-0 SCKSET	SCK Setup Configures the setup phase of the SCK pin in Controller mode:

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	• The setup phase is the delay between the SCK edge that drives the transmit data and the SCK edge that samples the receive data. • It is the SCK high period when CPHA = 0 and CPOL = 1, or CPHA = 1 and CPOL = 0. It is the SCK low period when CPHA = 0 and CPOL = 0, or CPHA = 1 and CPOL = 1. • The SCK setup phase delay is equal to (SCKSET + 1) cycles of the LPSPI functional clock divided by the selected prescaler (see TCR[PRESCALE]). • The minimum delay is one cycle. • The SCK period is equal to (SCKSET + SCKHLD + 2) cycles of the LPSPI functional clock divided by the selected prescaler (see TCR[PRESCALE]). • The SCK duty cycle is based on the difference between SCKSET and SCKHLD. You must configure both these fields to the same value for a 50/50 duty cycle. See Figure 140 for more information. **NOTE** For backward compatibility, writing to CCR[SCKDIV] updates this field with (SCKDIV ÷ 2) rounded up.

33.6.1.14 FIFO Control (FCR)

Offset

Register	Offset
FCR	58h

Function

Contains the receive FIFO and transmit FIFO watermark values.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0								0						RXWATER	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								0						TXWATER	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Fields

Field	Function
31-24 —	Reserved
23-18 —	Reserved
17-16 RXWATER	Receive FIFO Watermark Causes LPSPI to set SR[RDF] when the number of words in the receive FIFO is greater than RXWATER. Writing a value equal to or greater than the FIFO size truncates the written value.
15-8 —	Reserved
7-2 —	Reserved
1-0 TXWATER	Transmit FIFO Watermark Causes LPSPI to set SR[TDF] when the number of words in the transmit FIFO is equal to or less than TXWATER. Writing a value equal to or greater than the FIFO size truncates the written value.

33.6.1.15 FIFO Status (FSR)

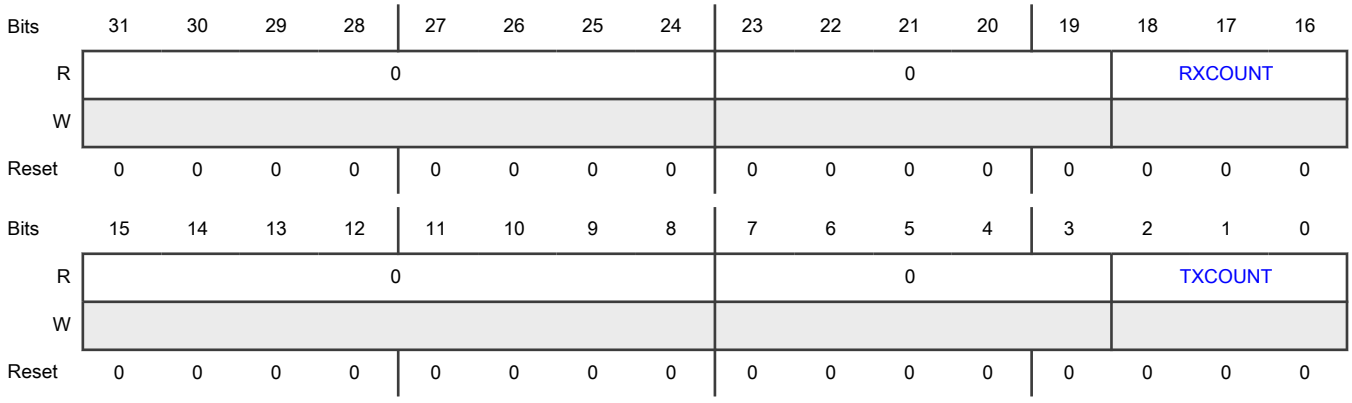
Offset

Register	Offset
FSR	5Ch

Function

Contains fields that indicate the number of words currently stored in the receive and transmit FIFOs.

Diagram



All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Fields

Field	Function
31-24 —	Reserved
23-19 —	Reserved
18-16 RXCOUNT	Receive FIFO Count Indicates the number of words currently stored in the receive FIFO.
15-8 —	Reserved
7-3 —	Reserved
2-0 TXCOUNT	Transmit FIFO Count Indicates the number of words currently stored in the transmit FIFO.

33.6.1.16 Transmit Command (TCR)

Offset

Register	Offset
TCR	60h

Function

Pushes the data into the transmit FIFO, in the same order as written.

When you write to either this register or to Transmit Data (TDR), each write pushes data into the transmit FIFO. You must write to this register only using 32-bit writes, which are tagged and cause the command register to update; after that the entry reaches the top of the FIFO and LPSPI is enabled. This allows changes to the command word and the transmit data itself to be interleaved. That is, writes to the two registers can be interleaved (write command word, then data word, then command word, and so on). Changing the command word causes all subsequent SPI bus transfers to be performed using the new command word:

- In Controller mode, writing a new command word does not initiate a new transfer, unless TXMSK is 1. Transfers are initiated by transmit data in the transmit FIFO, or by a new command word (with TXMSK = 1). Hardware writes 0 to TXMSK when PCS deasserts.
- In Controller mode, if the command word is changed before an existing frame has completed, then the existing frame terminates and the command word updates. The command word can be changed during a continuous transfer, if CONTC of the new command word is 1 and the command word is written on a frame size boundary.
- In Peripheral mode, the command word must be changed only when LPSPI is idle and there is no SPI bus transfer.

Avoid resetting the transmit FIFO after writing to this register; wait for the command register to update from the FIFO first.

Avoid register reading problems: Reading this register returns the current state of the register. Reading this register at the same time that it is loaded from the transmit FIFO can return an incorrect register value. It is recommended to:

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

- Read this register when the transmit FIFO is empty.
- Read this register more than once and then compare the returned values.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	CPOL	CPHA	PRESCALE			Reserved	PCS		LSBF	BYSW	CONT	CONTC	RXMSK	TXMSK	WIDTH	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved				FRAMESZ											
W	0															
Reset	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1

Fields

Field	Function
31 CPOL	Clock Polarity Specifies the value of SCK when it is idle. You can update this field only when PCS is deasserted. See Figure 140 for more information. 0b - Inactive low 1b - Inactive high
30 CPHA	Clock Phase Indicates whether data is captured or changed on the leading edge of SCK and captured or changed on the following edge of SCK. You can update this field only when PCS is deasserted. See Figure 140 for more information. 0b - Captured 1b - Changed
29-27 PRESCALE	Prescaler Value Specifies the division of the LPSPI functional clock. For all SPI bus transfers, this value is applied to Clock Configuration (CCR). You can update this field only when PCS is deasserted. 000b - Divide by 1 001b - Divide by 2 010b - Divide by 4 011b - Divide by 8 100b - Divide by 16 101b - Divide by 32

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	110b - Divide by 64 111b - Divide by 128
26 —	Reserved
25-24 PCS	Peripheral Chip Select Configures the peripheral chip select used for the transfer. This field is updated only when PCS is deasserted. **NOTE** This entire field is not fully supported in every LPSPI module instance. See the chip-specific LPSPI information. 00b - Transfer using PCS[0] 01b - Transfer using PCS[1] 10b - Transfer using PCS[2] 11b - Transfer using PCS[3]
23 LSBF	LSB First Indicates whether data is transferred with MSB first or LSB first. 0b - MSB first 1b - LSB first
22 BYSW	Byte Swap Swaps the contents of [31:24] with [7:0] and [23:16] with [15:8] for each transmit data word read from the FIFO and for each received data word stored to the FIFO (or compared with match registers). 0b - Disable byte swap 1b - Enable byte swap
21 CONT	Continuous Transfer Enables continuous transfer: • In Controller mode, this field keeps PCS asserted at the end of the frame size until a command word is received that starts a new frame. • In Peripheral mode, when this field is enabled, LPSPI only transmits the first FRAMESZ bits, after which LPSPI transmits received data (assuming a 32-bit shift register) until the next PCS negation. 0b - Disable 1b - Enable
20 CONTC	Continuing Command Enables the command word to be changed within a continuous transfer in Controller mode:

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	• The initial command word must enable continuous transfer (CONT = 1). • The continuing command must have CONTC = 1. • The continuing command word must be loaded on a frame size boundary. For example, if the continuous transfer has a frame size of 64 bits, then a continuing command word must be loaded on a 64-bit boundary. In Peripheral mode, this field modifies the internal RXMSK and TXMSK configuration after the first FRAMESZ bits and until PCS negation: • Receive data is discarded after the first FRAMESZ bits. If CONT is also 1, this does not block the transmission of received data. • Transmit data is not masked after the first FRAMESZ bits. This allows the first FRAMESZ bits to be received and a response transmitted. 0b - Command word for start of new transfer 1b - Command word for continuing transfer
19 RXMSK	Receive Data Mask Masks receive data (receive data is not stored in the receive FIFO). 0b - Normal transfer 1b - Mask receive data
18 TXMSK	Transmit Data Mask Masks transmit data (no data is loaded from the transmit FIFO and the output pin is 3-stated). In Controller mode, TXMSK initiates a new transfer that cannot be aborted by another command word. TXMSK automatically transitions to 0 at the end of the transfer. 0b - Normal transfer 1b - Mask transmit data
17-16 WIDTH	Transfer Width Configures serial (1-bit) or parallel transfers. For half-duplex parallel transfers, either RXMSK or TXMSK must be 1. 00b - 1-bit transfer 01b - 2-bit transfer 10b - 4-bit transfer 11b - Reserved
15-12 —	Reserved
11-0 FRAMESZ	Frame Size Configures the frame size in number of bits equal to (FRAMESZ + 1):

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	• The minimum frame size is 8 bits. • If the frame size is larger than 32 bits, then the frame is divided into multiple words of 32 bits; each word is loaded from the transmit FIFO and stored in the receive FIFO separately. • If the size of the frame is not divisible by 32, then the last load of the transmit FIFO and store of the receive FIFO contains the remainder bits. For example, a 72-bit transfer consists of three words: the first and second words are 32 bits, and the third word is 8 bits.

33.6.1.17 Transmit Data (TDR)

Offset

Register	Offset
TDR	64h

Function

Pushes the data into the transmit FIFO, in the same order that the data is written. You can write to this register using 32-, 16-, or 8-bit writes.

When you write to this register or to Transmit Command (TCR), each write pushes data into the FIFO with zero pushed in unwritten bytes.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R																
W	DATA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R																
W	DATA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-0 DATA	Transmit Data Indicates transmit data. Both 8-bit and 16-bit writes of transmit data zero-extend the data written and push the data into the transmit FIFO. To zero-extend 8-bit and 16-bit writes (to 32 bits) means that the higher order (most significant) empty parts of the 8-bit and 16-bit writes are filled with zeroes.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

33.6.1.18 Receive Status (RSR)

Offset

Register	Offset
RSR	70h

Function

Contains data flow status fields for receive FIFO.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0														RXEMPTY	SOF
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

Fields

Field	Function
31-2 —	Reserved
1 RXEMPTY	RX FIFO Empty Indicates whether the receive FIFO is empty. 0b - Not empty 1b - Empty
0 SOF	Start of Frame Indicates whether this is the first data word received after PCS assertion. 0b - Subsequent data word or RX FIFO is empty (RXEMPTY=1). 1b - First data word

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

33.6.1.19 Receive Data (RDR)

Offset

Register	Offset
RDR	74h

Function
Pulls the first entry from the receive FIFO.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	DATA															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DATA															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-0 DATA	Receive Data

33.6.1.20 Receive Data Read Only (RDROR)

Offset

Register	Offset
RDROR	78h

Function
Returns the first entry in the receive FIFO but does not remove the data from the FIFO.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	DATA															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DATA															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-0 DATA	Receive Data

33.6.1.21 Transmit Command Burst (TCBR)

Offset

Register	Offset
TCBR	3FCh

Function

Supports burst transfers of command data to the transmit FIFO for use with the DMA controller.

See DMA support registers.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R																
W	DATA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R																
W	DATA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Fields

Field	Function
31-0 DATA	Command Data Writes data to Transmit Command (TCR).

33.6.1.22 Transmit Data Burst (TDBR0 - TDBR127)

Offset

For n = 0 to 127:

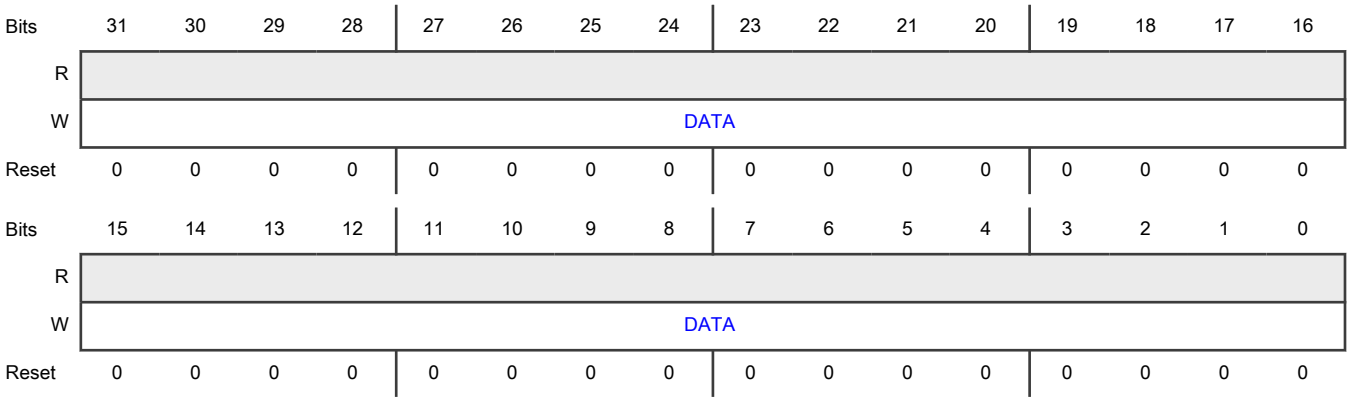
Register	Offset
TDBRn	400h + (n × 4h)

Function

Supports burst transfers of data to the transmit FIFO for use with the DMA controller. The size of this register is 512 bytes.

See DMA support registers.

Diagram



Fields

Field	Function
31-0 DATA	Data Writes data to Transmit Data (TDR).

33.6.1.23 Receive Data Burst (RDBR0 - RDBR127)

Offset

For n = 0 to 127:

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Register	Offset
RDBRn	600h + (n × 4h)

Function

Supports burst transfers of data from the receive FIFO. The size of this register is 512 bytes.

See DMA support registers.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	DATA															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DATA															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-0 DATA	Data Reads data from Receive Data (RDR).

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Chapter 34
Analog-to-Digital Converter (ADC)

34.1 Chip-specific ADC information

Table 206. Reference links to related information

Topic	Related module	Reference
Full description	ADC	ADC
System memory map		Memory map
Clocking		Clock distribution
Power Management		Power Management
Signal multiplexing	Port control	Signal multiplexing

34.1.1 Module instances

This chip contains one instance of 16-bit ADC: ADC0.

34.1.2 ADC input connections

NOTE

Not all the pins are available for each package, see Introduction for more details.

Table 207. ADC analog channel input connections

ADC channel (CMDLn[ADCH])	ADC0	Description
0	ADC0_A0 (P2_0)	VDD domain, 40 Ohm
1	ADC0_A1 (P2_1)	VDD domain, 40 Ohm
2	ADC0_A2/OPAMP0_EXT_REF1 (P2_3)	VDD domain, 40 Ohm
3	ADC0_A3/OPAMP0_INT	—
4	ADC0_A4/CMP0_IN0 (P2_2)	VDD domain, 40 Ohm
5	ADC0_A5/OPAMP0_INP0 (P2_12)	VDD domain, 40 Ohm
6	ADC0_A6/OPAMP0_OUT/CMP0_IN4P (P2_16)	VDD domain, 40 Ohm
7	VREFI/ADC0_A7/OPAMP0_EXT_REF0	VDD domain, 40 Ohm

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.
© 2026 NXP B.V. All rights reserved.

Table 207. ADC analog channel input connections (continued)

ADC channel (CMDLn[ADCH])	ADC0	Description
	(P2_7)	
8	ADC0_A8 (P1_10)	VDD domain, analog mux
9	ADC0_A9 (P1_11)	VDD domain, analog mux
10	Reserved	—
11	Reserved	—
12	ADC0_A12 (P3_31)	VDD domain, analog mux
13	ADC0_A13 (P3_30)	VDD domain, analog mux
14	ADC0_A14 (P3_29)	VDD domain, analog mux
15	ADC0_A15 (P0_6)	VDD domain, analog mux
16	ADC0_A16/CMP0_IN3 (P1_0)	VDD domain, analog mux
17	ADC0_A17 (P1_1)	VDD domain, analog mux
18	ADC0_A18 (P1_2)	VDD domain, analog mux
19	ADC0_A19 (P1_3)	VDD domain, analog mux
20	Reserved	—
21	Reserved	—
22	Reserved	—
23	Reserved	—
24	VSSA	—
25	Reserved	—
26	Temperature+	—

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table 207. ADC analog channel input connections (continued)

ADC channel (CMDLn[ADCH])	ADC0	Description
27	PMC BG	ADC internal PMC bandgap, pmc_1vbuf_ana_1p8v
28	OPAMP0_OBS	—
29	VDDA/4	PMC resistive dividers

34.1.3 ADC voltage reference options

The ADC voltage references can be selected by CFG[REFSEL] as follows:

- CFG[REFSEL]=00, VDDA_ANA supply pin
- CFG[REFSEL]=01, VREFI
- CFG[REFSEL]=10, VDDA_ANA supply pin

34.1.4 ADC trigger inputs

ADC trigger sources get routed through the Input Multiplexing (INPUTMUX). See the INPUTMUX chapter for available trigger sources.

34.2 Overview

16-bit ADC is a successive approximation ADC designed for operation within an integrated microcontroller system-on-chip.

NOTE

For the chip specific modes of operation, see the power management information of the device.

34.2.1 Block diagram

The following figure is the ADC module block diagram.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

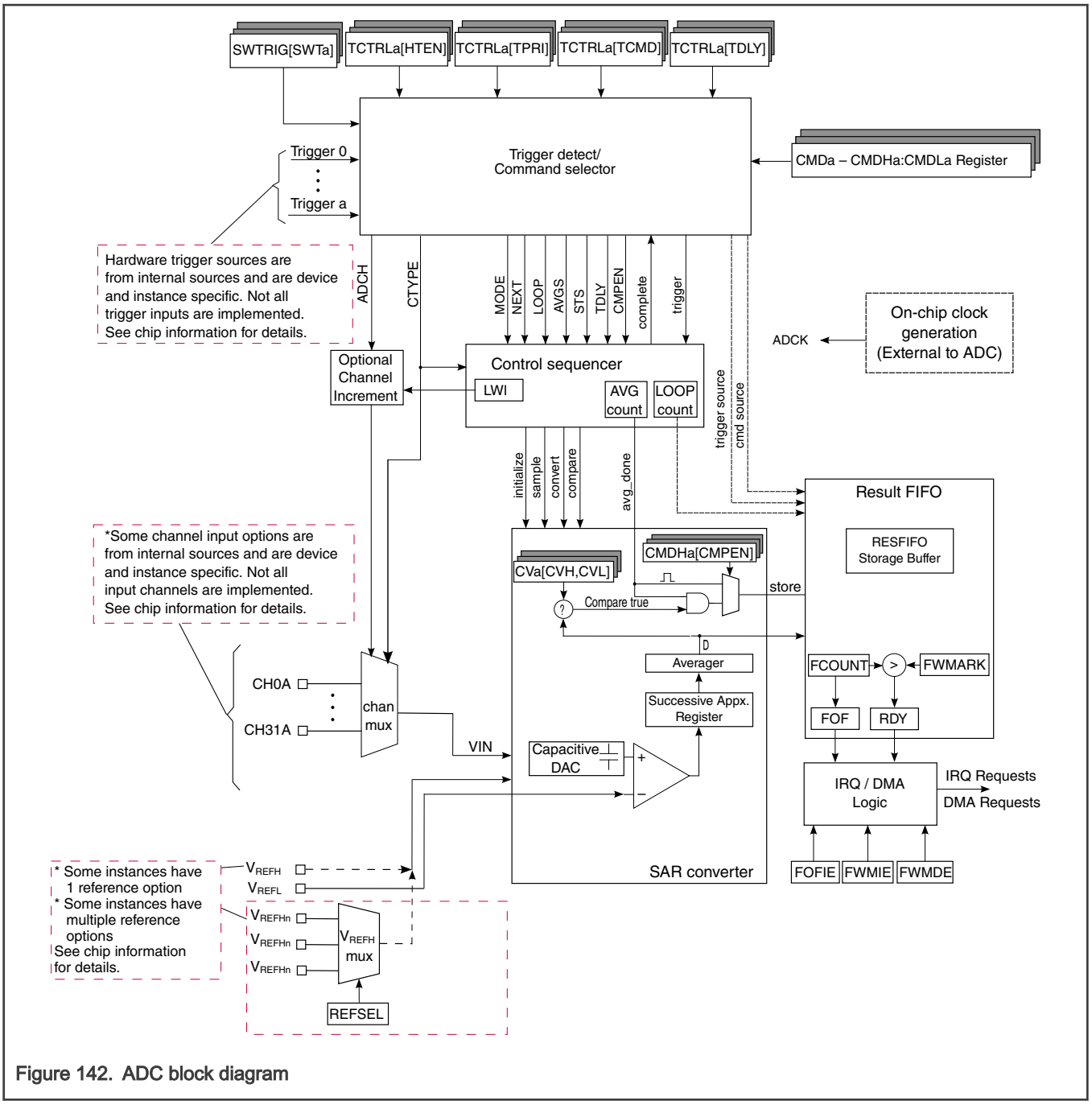


Figure 142. ADC block diagram

34.2.2 Features

Following are the features of the ADC module.

- Linear successive approximation algorithm
  - single-ended operation with 12-bit resolution or reduced bandwidth 16-bit resolution.
- Configurable analog input sample time
- Configurable speed options to accommodate operation in low power modes of SoC
- Trigger detect with up to 4 trigger sources with priority level configuration. Software or hardware trigger option for each.
- 7 command buffers allow independent options selection and channel sequence scanning.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

- Automatic compare for less-than, greater-than, within range, or out-of-range with "store on true" and "repeat until true" options
- 8-entry conversion result data FIFO with configurable watermark and overflow detection
- Interrupt, DMA or polled operation
- Linearity and gain adjustment calibration logic

34.3 Functional description

The ADC module performs analog-to-digital conversions on any of the software selectable analog input channels by a successive approximation algorithm. The module initializes to its lowest power state during reset. The ADC analog circuits can optionally be pre-enabled for faster starts to conversions at the expense of higher idle currents. Conversions are initiated by selectable trigger events from software or hardware sources. The trigger detect logic includes a configurable enable and priority scheme for the available trigger sources. The module includes multiple command buffers that provide configurable flexibility for channel scanning and independent channel selections for different trigger sources. Multiple command buffers also allow variable option selection such as sample time and averaging on a per-channel basis.

The ADC module optionally averages the result of multiple conversions on a channel before storing the calculated result. The hardware average function is enabled by setting CMDH*a*[AVGS] bitfield to a non-zero value and operates in any of the conversion modes and configurations.

When the conversion and averaging loops are completed, the resulting data is placed in a FIFO data buffer along with other tag information associated with the result. A configurable watermark level supports interrupt or DMA requests when the number of stored datawords exceeds the setting. Interrupts can also be enabled to indicate when FIFO overflow errors occur.

The ADC module optionally compares the result of a conversion with the contents of two value registers for less-than, greater-than, inside-range or outside-range detection. The compare function operates in any of the conversion modes and configurations.

The ADC module includes offset and linearity calibration logic. A request for calibration should be made any time upon reset or power up. Each SAR conversion utilizes calibration data calculated during the calibration routine.

The sequencing of a ADC command is summarized in the flow diagram shown below.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

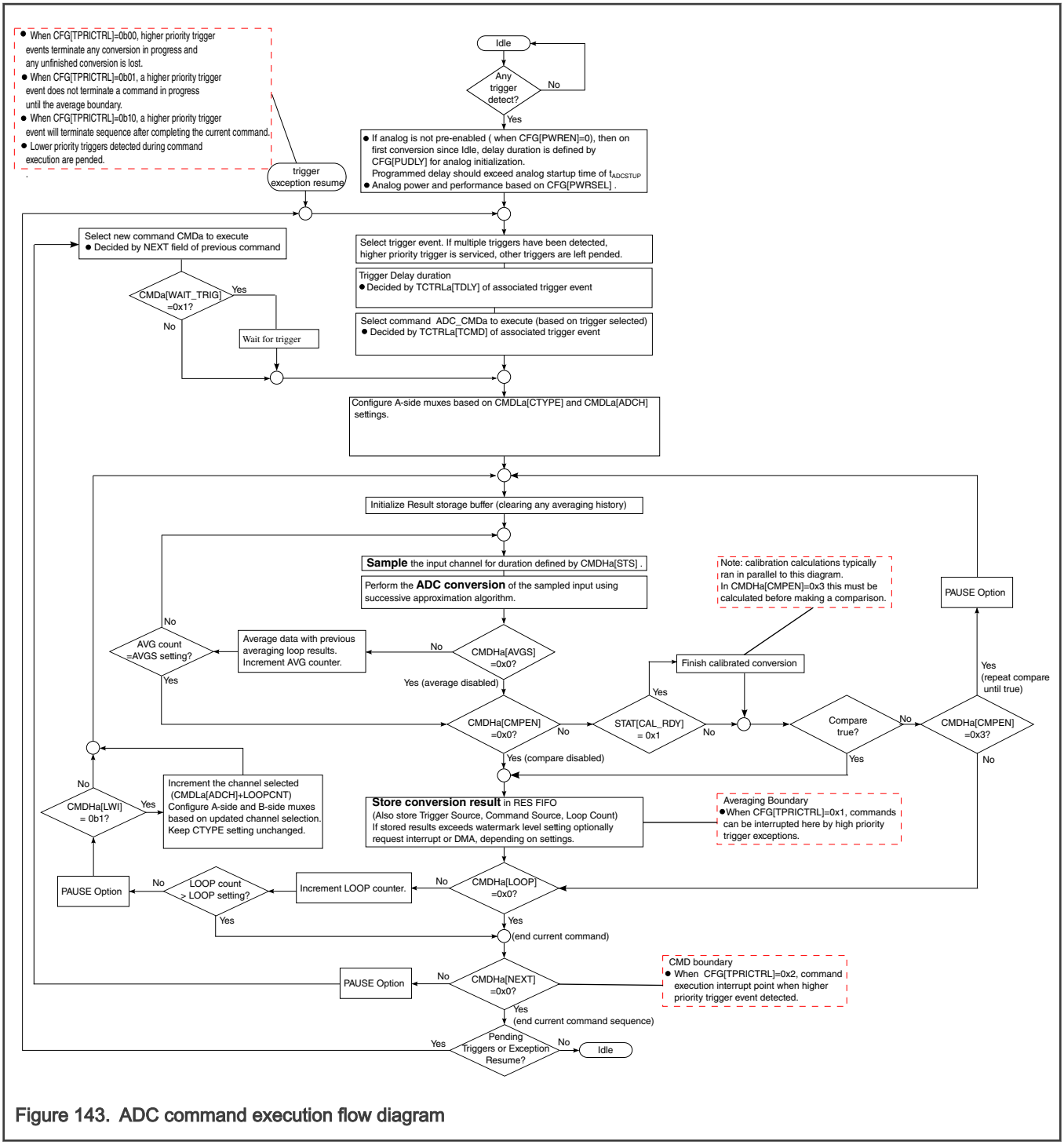


Figure 143. ADC command execution flow diagram

34.3.1 Power control mode

By default, the ADC analog circuits are disabled while ADC is in its idle state. When a trigger is detected and ADC command processing is initiated, the analog circuits are enabled. These circuits require a period of initialization before the first conversion cycle.

The value of CFG[PUDLY] should be set to incur a delay longer than $t_{ADCSTUP}$. The accuracy of initial conversions after activation is degraded if the value of CFG[PUDLY] is too low.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

You can achieve faster conversion startup times by setting CFG[PWREN] to pre-enable the analog circuits of ADC. This faster conversion consumes extra power, even while ADC is in an idle state. When CFG[PWREN] is 1, the Power Enable timer is activated. The timer enforces the minimum time required (configured by CFG[PUDLY]) before detected triggers can initiate ADC conversions.

ADC also has options for controlling power and performance summarized in the table below. See the device data sheet for specification of the available power modes.

Table 208. Power option settings

CFG[PWRSEL]	Description
0b (Default)	Slow speed and low power
1b	Fast speed and high power

34.3.2 ADC behavior in low power mode

The ADC module supports system level low power modes. In system low power modes where the ADC can remain functional, the CTRL[DOZEN] bit controls ADC behavior.

With the CTRL[DOZEN] bit clear, if the ADC was configured and enabled before low power mode entry, ADC is allowed to continue operation while the system is transitioned to the low power state. Any conversion in progress is not disrupted. External hardware trigger detect and active conversions are operational.

With the CTRL[DOZEN] bit set the ADC waits for the current averaging iteration/FIFO storage to complete before acknowledging low power entry request. After system entry to the low power state, the ADC is kept in inactive state until the system exits the low power state.

When entering a low power mode where the ADC cannot remain functional, the CTRL[DOZEN] bit is ignored and the ADC always waits for the current averaging iteration/FIFO storage to complete before acknowledging the low power entry request. After entry to the low power state, the ADC is kept in an inactive state until the system exits the low power state. The CFG[PWREN] bit setting is ignored and the ADC analog is forced to its lowest power state. Upon wakeup from the low power state, if the analog was pre-enabled before stop mode entry (i.e., the CFG[PWREN] bit is set) and the control registers have retained state in the low power mode, the analog is automatically re-enabled and the power up delay timer begins. Any triggered conversion is stalled until the power up delay associated with the CFG[PUDLY] control field has finished.

34.3.3 Voltage reference

The voltage reference high (V_{REFH}) used by the ADC is supplied from either an on-chip voltage reference source or from an off-chip source supplied through external pins. V_{REFL} is always from an external pin and must be at the same voltage potential as V_{SSA}. See the chip configuration information on the voltage reference options specific to this packaged device.

This block supports a programmable selection of the Voltage Reference High used for ADC conversions (via the CFG[REFSEL] field). See the chip configuration information on the voltage reference options specific to this packaged device.

34.3.4 Trigger detect and command execution

See Figure 143 for a flow diagram of command execution sequencing.

ADC command execution is initiated from up to 4 trigger sources. Each trigger can be software generated by writing 0b1 to the corresponding SWTRIG[SWT*n*] bitfield. Alternatively, hardware triggers can be generated from asynchronous input sources at the periphery of the module. The number and sources of hardware triggers implemented is device specific. Refer to chip configuration information for description of available hardware trigger sources for this device.

Each hardware trigger source is enabled by setting the associated enable bit (TCTRL*a*[HTEN]). Each trigger source is assigned a priority via the associated priority control field (TCTRL*a*[TPRI]). Each of the trigger sources is associated with a command buffer via the associated command select field (TCTRL*a*[TCMD]).

When a hardware trigger input is enabled, hardware trigger events are detected on the rising-edge of the associated hardware trigger source.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

When the ADC is inactive, achieving the minimum startup time to begin an initial conversion after a software or hardware generated trigger requires that the ADC be pre-enabled (by setting CFG[PWREN] during ADC initialization and allowing enough time for ADC power-up delay to timeout before allowing any triggers to be generated). In the case of a pre-enabled ADC, it takes 9 ADCK cycles for the trigger to be recognized and begin sampling on the associated analog channel.

In the case that the ADC is not pre-enabled, an initial conversion is delayed further as determined by the power-up delay (CFG[PUDLY]). Conversion startup can also be delayed on a trigger source basis (TCTRLn[TDLY]).

Each trigger source has an associated priority field TCTRLa[TPRI] which allows for arbitration between trigger sources. Arbitration is in control of two things: selecting which trigger sequence to execute next, and selecting how to handle a trigger exception. Trigger exceptions are defined as allowing a higher priority trigger sequence to interrupt operation of a lower priority sequence. When a trigger exception occurs, programmable arbitration allows the configurable stop and resume points for low priority sequences. The fields affecting arbitration are CFG[HPT_EXDI],CFG[TCMDRES],CFG[TRES], and CFG[TPRICTRL].

1. If CFG[HPT_EXDI] is set to 1'b1 then trigger exceptions are disabled and any higher priority triggers are left pending until the current sequence completes. Note that new triggers are accepted based on priority.
2. If CFG[HPT_EXDI] is set to 1'b0 (default), then exceptions are enabled and the higher priority sequence begins executing at a user specified breakpoint.

Breakpoint locations are determined by the register CFG[TPRICTRL]. CFG[TPRICTRL] has an affect on latency for accepting a trigger exception.

1. When CFG[TPRICTRL]=0x0, a higher priority trigger causes an immediate command abort and the new command specified by the trigger is immediately started. It takes 9 ADCK cycles for the higher priority trigger to be recognized, terminate the conversion in progress, and complete a switch to the new associated ADC command. 9 ADCK cycles after the trigger is driven to the ADC, the ADC begins sampling on the associated new analog channel.
2. When CFG[TPRICTRL]=0x1, the current conversion is allowed to complete (including averaging) before the higher priority exception is initiated. In this mode, if the command is running through a series of averages, this series completes. However, there is no requirement to finish the entire command before being interrupted. For example, if the command consists of 4 loop iterations, there is no requirement to complete all 4 iterations before the interrupt occurs.
3. When CFG[TPRICTRL]=0x2, a higher priority trigger begins once the current command is completed. If a command consists of 5 loop iterations each containing 8 averages, then all 5x8 conversions must be completed before accepting the trigger exception.

CFG[TCMDRES] and CFG[TRES] determine what the ADC does after accepting a trigger exception. The module can be programmed to resume commands after returning from a trigger exception.

1. If CFG[TRES] = 0x0 then commands are not automatically resumed after being stopped by an exception. However, an interrupt is set to indicate this case has occurred. The flag TSTAT[TEXC_NUM] can be used to resolve which trigger was stopped by the exception.
2. If CFG[TRES] = 0x1 the ADC automatically resumes commands after they were stopped by an exception.

By utilizing CFG[TRES] in conjunction with CFG[TCMDRES], the module can be programmed to resume commands at one of two possible locations.

1. If CFG[TCMDRES] = 0x0 then the trigger which was stopped by an exception is resumed from the beginning of its associated command sequence. Note, triggers which are waiting to be resumed take the same priority programmed to TCTRLa[TPRI].
2. If CFG[TCMDRES] = 0x1 then the trigger is resumed from the command that it was executing before being interrupted by an exception.

If a lower priority trigger occurs (i.e., a trigger event occurs that is configured for a lower priority than the trigger source associated with the currently executing command), the trigger detect is left pending until completion of the current command sequence. Lower priority trigger events cannot be serviced until a higher priority triggered command (or command sequence) completes.

When a conversion is completed (including hardware averaging when CMDHa[AVGS] is non-zero), the result is placed in a RESFIFO buffer. When an ADC command selects looping (when CMDHa[LOOP] is non-zero) a command stores multiple conversion results to the FIFO during execution of that command.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

At the end of command execution, the CMDHa[NEXT] field of the command selects the next command to be executed. Multiple commands can be executed sequentially by configuration of each commands CMDHa[NEXT] field. Setting the next command to 0x0 causes conversions to terminate at the completion of the command. Unending circular command execution is allowed by setting the CMDHa[NEXT] field in the last command in a sequence to the first command in the sequence.

By default, command sequences executes automatically in the order that CMDHa[NEXT] fields are programmed. However, by utilizing the CMDHa[WAIT_TRIG], command execution can be stalled and launched based on trigger inputs. For example, if TRIGGER2 is programmed to start the command sequence CMD1, CMD2, CMD3, then receiving TRIGGER 2 one time unconditionally runs this sequence to completion. If CMDH2[WAIT_TRIG] is set to 0x1, however, then the sequence pauses after CMD1 until TRIGGER2 is received again. Therefore, sequences can be stalled until receiving a trigger assertion.

Disabling the ADC by writing 0b0 to the CTRL[ADCEN] bitfield terminates any active ADC command processing. Writing 0b0 to the CTRL[ADCEN] bitfield causes the current command (or command sequence) to terminate, clears any pending triggers and sends the ADC module to an IDLE state.

34.3.5 Pause option

When the maximum conversion rate is not required by an application the effective conversion rate can be reduced by implementing periodic trigger events to initiate ADC conversions or by selecting a reduced frequency clock as the ADACK source. Both of these options are chip specific and are dependent on ADC triggering and clocking options external to the ADC module. The latency associated with ADC analog power up delays results in a limit on the maximum conversion rate when using periodic triggering.

Another means of reducing conversion rates is by inserting a pause of a programmable duration between LOOP iterations, between commands in a sequence, and between conversions when command is executing in the "Compare Until True" configuration. When PAUSE[PAUSEEN] is set, the PAUSE[PAUSEDLY] field controls the duration of pausing during command execution sequencing. The pause delay is a count of (PAUSE[PAUSEDLY]*4) ADCK cycles. Note, the PAUSE register should not be changed while the CTRL[ADCEN] bit is set. Writes to the PAUSE register while CTRL[ADCEN] is set can lead to metastable operation.

See Figure 143 for the places during command execution sequencing where the pause is optionally inserted.

34.3.6 Resync functionality

Any trigger source (SW or HW) can be configured to act as a resync trigger. Trigger based resync functionality is used to interrupt a running trigger (resync target) and clear the FIFO it's writing to. This can either be used to abort a running sequence, or restart a running sequence depending on the configuration of CFG[TRES]. If CFG[TRES] = 0b1 then the target sequence is aborted, the FIFO cleared, and the sequence restarted after the resync occurs. If CFG[TRES] = 0b0 then the target sequence is aborted and the FIFO is cleared after the RESYNC occurs.

A resync trigger needs to have a specific target. The resync only occurs if the resync target is running at the time of the trigger. For the following description, let n be the resync trigger number, let m be the resync target number. According to these variables, trigger n should resync trigger m. To enable a trigger source to act as a resync trigger, the following conditions must be satisfied:

1. The resync trigger TCTRLn[RSYNC] must be set to 0b1.
2. The resync trigger must have higher priority than the resync target (TCTRLn[TPRI] must be less than TCTRLm[TPRI]).
3. The resync target is specified using TCTRLn[TCMD]. In this case the resync target, m, must be equal to TCTRLn[TCMD].
4. The resync target, m, must be executing commands when the resync trigger, n, is asserted.
5. Trigger m must have at least one conversion left to begin when trigger n is received.

If a trigger source n has TCTRLn[RSYNC] set to 0b1, but some of the above conditions are not met then the trigger source n is ignored.

The following figure illustrates a resync trigger sequence executing. Note, in this example, trigger source 1 is configured to resync trigger source 0.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

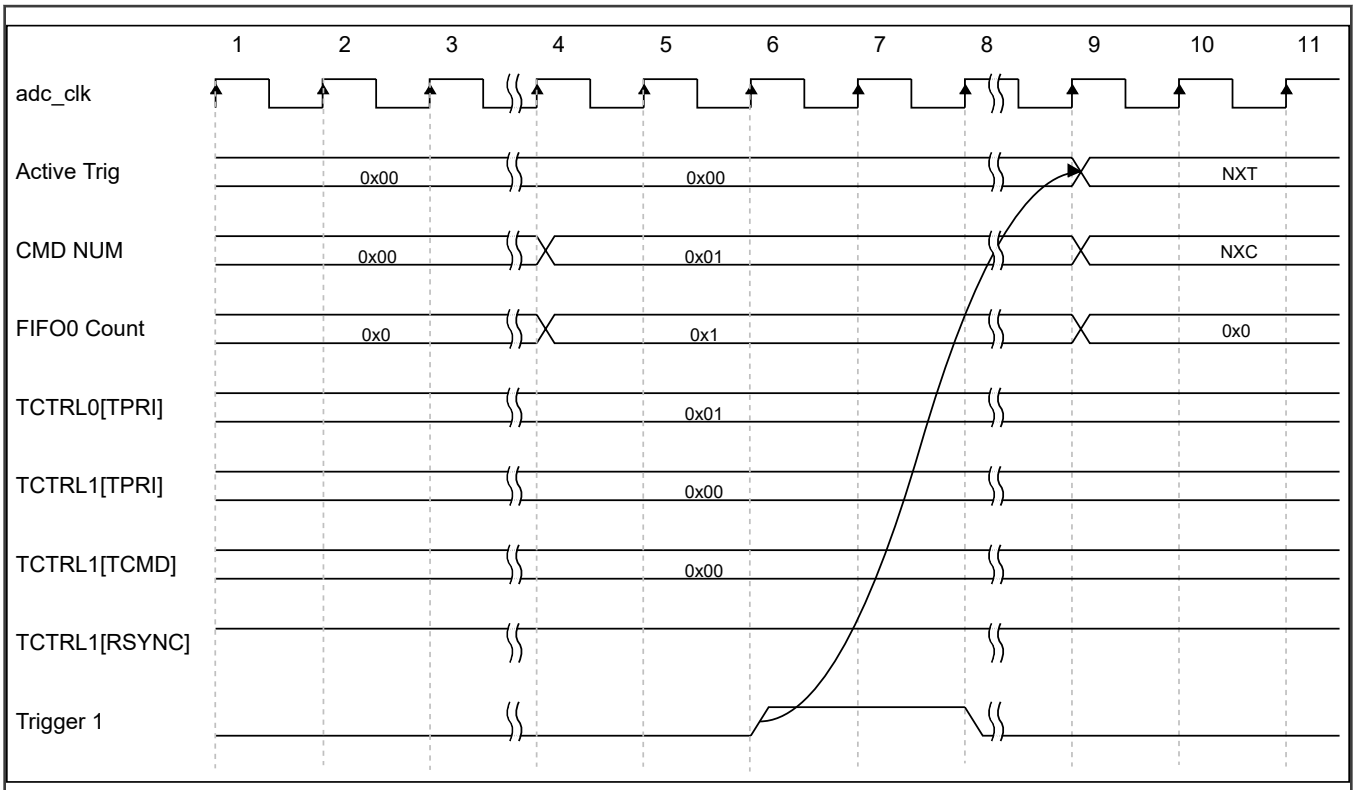


Figure 144. ADC Resync Example

In this figure, trigger source 0 was executing a sequence of commands when trigger source 1 was asserted. Notice that trigger source 0 is stopped when trigger source 1 is asserted (after some synchronization delay). In addition, the FIFO being written to by trigger source 0 is cleared (FIFO0 in this example). After the trigger 0 sequence is stopped, the ADC runs the next trigger pending with the highest priority. This is marked as NXT, for next trigger. If resume functionality is enabled, and trigger source 0 had the highest priority pending, then NXT = 0x00.

34.3.7 Temperature sensor

The ADC module has a dedicated input channel for an on-chip temperature sensor. Refer to the chip specific channel definition to determine which channel is connected to the on-chip temperature sensor.

To calculate the temperature, application software must first execute a conversion of the temperature sensor channel with some configuration requirements. This sequence for converting the temperature sensor channel is outlined below:

1. Refer to chip specific channel definition to determine which channel corresponds to the temperature sensor.
2. Configure a command buffer register to convert the temperature sensor channel. CMDLa[ADCH] = Temperature Sensor Channel.
3. The command buffer must also be programmed with the following parameters:
   - 16b mode (CMDLa[MODE] = 0x1)
   - Max averaging (CMDHa[AVGS] = 0xA)
   - Max sample time (CMDHa[STS] = 0x7)
   - LOOP set to 1 (CMDHa[LOOP] = 0x1)
   - Loop with increment disabled (CMDHa[LWI] = 0x0)
   - Compare function disabled (CMDHa[CMPEN] = 0x0)
4. Configure a trigger control register TCTRLa with the following parameters:

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

- TCTRLa[TCMD] = command buffer used in steps 2 and 3.

5. Trigger a conversion of the temperature sensor channel. Software triggers the conversion by writing "1" to the associated bit in SWTRIG register (i.e., the trigger configured in step 4).

After completion of the conversion of the temperature sensor channel, two results are stored in the FIFO. Each result corresponds to a component of the overall temperature value. Application software reads these values from the FIFO and uses the equation below to calculate the ambient temperature.

$$Temp = A\left[\frac{\alpha(Vbe8 - Vbe1)}{Vbe8 + \alpha(Vbe8 - Vbe1)}\right] - B$$

Equation 3. Temperature sensor function

Where:

- Vbe1 is the first value stored to the FIFO as a result of the temperature sensor channel conversion
- Vbe8 is the second value stored to the FIFO as a result of the temperature sensor channel conversion
- A is the slope factor
- B is the offset factor
- α is the bandgap coefficient

A, B, and α are specified constant values from the ADC Electrical information in the device datasheet.

34.3.8 Result FIFO operation

The ADC includes 1 8-entry FIFO in which the result of ADC conversions are stored. In addition, a valid indicator bit, the trigger source, the source command and the loop count are also stored along with the data. FCTRL0[FCOUNT] indicates how many valid datawords are stored in each RESFIFO.

A programmable watermark threshold supports configurable notification of data availability. When FCTRL0[FCOUNT] is greater than FCTRL0[FWMARK], the associated RDY flag is asserted. When IE[FWMIE0] is set, a watermark interrupt request is issued. When DE[FWMDE0] is set, a DMA request is issued. Reading RESFIFO provides the oldest unread dataword entry in the FIFO and decrements FCTRL0[FCOUNT]. When FCTRL0[FCOUNT] falls equal to or below FCTRL0[FWMARK], the RDY flag is cleared.

Each FIFO can be emptied by successive reads of RESFIFO0. When the RESFIFO0[VALID] bit is 1 the associated FIFO entry is valid. Reading RESFIFO0 when the FIFO is empty (when RESFIFO0[VALID] is clear and FCTRL0[FCOUNT]=0x0) provides an undefined dataword. All FIFOs are reset by writing 0b1 to the CTRL[RSTFIFO0] bit.

If the ADC attempts to store a dataword to the FIFO when the FIFO is full the FIFO overflow flag (FCTRL0[FOF]) is set. When IE[FOFIE0] is set, a overflow interrupt request is issued. The FOF flag is cleared by writing 1 to STAT[FOF0]. On overflow events no new data is stored and the data associated with the store that triggered the overflow is lost.

34.3.9 Compare function

After the input is sampled and converted and any averaging iterations are performed, the CMDH*a*[CMPEN] field guides operation of the automatic compare function to optionally only store when the compare operation is true. There are multiple options on command sequencing related to the compare function as summarized in the table below.

NOTE

Latency is added to the end of a compare until true conversion to resolve the next command or loop in a sequence. This latency is necessary to calibrate the SAR data before resolving the result of a comparison. Delay for this feature is only added when resolving the result of a conversion. This means that intermediate samples during averaging do not include extra latency. Only loop and command boundaries experience this delay. The latency is always less than or equal to 5 ADC clock cycles.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

NOTE

Not all Command Buffers have an associated Compare Value register. The compare function is only available on Command Buffers that have a corresponding Compare Value register.

Table 209. Compare modes

CMDH*a*[CMPEN]	Compare Function	Description
0b00	Compare disabled	Do not perform compare operation. Always store the conversion result to the FIFO.
0b01	Reserved	
0b10	Store on true	Perform compare operation. Store conversion result to FIFO at end of averaging only if compare is true. If compare is false do not store the result to the FIFO. In either the true or false condition, the LOOP setting is considered and increments the LOOP counter before deciding whether the current command has completed or additional LOOP iterations are required.
0b11	Repeat compare until true	Perform compare operation. Store conversion result to FIFO at end of averaging only if compare is true. Once the true condition is found the LOOP setting is considered and increments the LOOP counter before deciding whether the current command has completed or additional LOOP iterations are required. If the compare is false do not store the result to the FIFO. The conversion is repeated without consideration of LOOP setting and does not increment the LOOP counter.

Depending on CV*a*[CVH] and CV*a*[CVL] values programmed, the compare operation checks whether the result is less than, greater than, or if the result falls within or outside a range determined by two compare values. The compare values are used as described in the following table.

Table 210. Compare operations

CV*a*[CVL] vs. CV*a*[CVH]	Operation	Description
set CVa[CVL] < CVa[CVH]	Outside range (General form)	Compare true if the result is less than CVa[CVL] value **OR** greater than CVa[CVH] value.
set CVa[CVH] to max value set CVa[CVL] to compare point	Less than	Compare true if the result is less than CVa[CVL] value.
set CVa[CVL] to min value set CVa[CVH] to compare point	Greater than	Compare true if the result is greater than CVa[CVH] value.
set CVa[CVL] > CVa[CVH]	Inside range	Compare true if the result is less than CVa[CVL] value **AND** greater than CVa[CVH] value.

NOTE

In low power modes where the ADC continues to operate, the compare function can monitor the voltage and only wake the device when the compare condition is met.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

34.3.10 Cycles per conversion

The number of ADCK cycles needed to complete a conversion varies based on the selected resolution (CMDLa[MODE]), the sample time configured, and several other configuration options discussed below.

To calculate the cycle count, first, determine the Base Cycles Count from Table 211. Next, add the Sample Time Adder based on CMDHa[STS] setting as summarized in Table 212. If averaging is enabled (CMDHa[AVGS] does not equal to 0), then the total cycles (Base Cycles Count + Sample Time Adder) should be multiplied by the number of averages configured as summarized in Table 213.

$$CycleCount\ /\ Conversion = [(BaseCycleCount + SampleTimeAdder) * AverageMultiplier]$$

Equation 4. Cycle Count/Conversion

Note that there is latency associated with trigger detection and starting an initial conversion. There is additional latency associated with offset and gain error adjustment of a raw conversion result and storage of a final result to the FIFO. Due to pipe-lining of conversions when looping or command chaining is configured, the next conversion is immediately started while a raw conversion result is being adjusted and stored and thus the conversion rate is maximized.

The base cycles per conversion is variable depending on CMDLa[MODE], CFG2[HS], CFG2[HSEXTRA], and CFG2[TUNE] settings as summarized in the following table.

Table 211. Base cycles per conversion

CFG2[HS] (default 0)	CFG2[HSEXTRA] (default 0)	CFG2[TUNE] (default 01)	Base ADCK cycles/conversion[1]		Description
			16-bit CMDLa[MODE] = 1	12-bit CMDLa[MODE] = 0	
0	X	00	24	20	
0	X	01	23	19	Reset default
0	X	10	22	18[2]	
1	0	00	21	17	
1	0	01	20	16	
1	0	10	19	15[2]	
1	1	00	22	18	
1	1	01	21	17	
1	1	10	20	16[2]	

1. This cycle count includes min sample time of 3.5 ADCK cycles.
2. Setting CFG2[TUNE] to 10 in 12-bit mode can result in missing codes (i.e., some result codes may not be produced).

In addition to the base cycles per conversion, the configured sample time needs to be considered. The sample time adder is variable depending on CMDHa[STS] setting and is summarized in the following table:

Table 212. Sample time cycle adder

CMDHa[STS] (default 000)	Add ADCK cycles/conversion
000	0
001	2

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table 212. Sample time cycle adder (continued)

CMDHa[STS] (default 000)	Add ADCK cycles/conversion
010	4
011	8
100	16
101	32
110	64
111	128

Table 213. Averaging multiplier

CMDHa[AVGS] (default 0000)	Averaging multiplier
0000	1
0001	2
0010	4
0011	8
0100	16
0101	32
0110	64
0111	128
1000	256
1001	512
1010	1024

34.3.11 Clocking

ADC uses the ADCK clock input provided by an on-chip clock select block. It is used by the SAR conversion-control sequencing logic and the FIFO storage buffer. The ADCK frequency must be within the specified frequency range for ADCK, and varies based on CFG[PWRSEL]. See the device data sheet for supported frequency ranges.

ADC continues operating in low-power mode as long as CTRL[DOZEN] = 0 and the on-chip clock select block supplies an ADCK clock source.

NOTE

When in low-power mode with CTRL[DOZEN] = 0, the bus clock can be shut off, and asynchronous interrupts and DMA requests can be configured. ADC continues processing commands and writing data to the internal FIFO.

ADC has four sources for asynchronous interrupts during low-power mode: watermark, FIFO overflow, TCOMP, and TEXC. To enable them, configure IE[FWMIE0], IE[FOFIE0], IE[TCOMP_IE], and IE[TEXC_IE] before entering low-power mode.

When CTRL[DOZEN] = 1 in low-power mode, ADC waits for the current averaging iteration or FIFO storage to complete before acknowledging the low-power mode entry. Any pending triggers are dropped when a low-power mode request is made in this mode. ADC is forced into its lowest-power setting after acknowledging the low-power mode request.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

34.3.12 Resets

Table 214. ADC Resets

Reset source	Description
Chip reset	The logic and registers for the ADC are reset to their default state on a chip reset.
Software reset	The ADC implements a software reset bit in its Control Register. The CTRL[RST] bit resets all logic and registers to their default state, except for the CTRL, GCRn and CAL_GARn registers.
FIFO reset	The ADC implements write-only control that reset the FIFO0 (CTRL[RSTFIFO0]). After a FIFO is reset, that FIFO is empty.

34.3.13 Interrupts and DMA requests

The ADC includes several sources for interrupts and/or DMA requests. The table below summarizes these sources.

A programmable watermark threshold supports configurable notification of data availability and can generate either an interrupt exception or a DMA request. When FCTRL0[COUNT] is greater than FCTRL0[FWMARK], the associated RDYn flag is asserted. Masking for this exception is controlled by the IE[FWMIE0] and DE[FWMDE0] control bits. When RDYn is asserted and mask control bit IE[FWMIE0] is set, a watermark interrupt request is issued. When RDYn is asserted and mask control bit DE[FWMDE0] is set, a DMA request is issued.

The other exception sources can only generate interrupts and do not have a DMA request option. Each of these sources has a mask control bit in the IE register and no corresponding bit in the DE register.

Table 215. ADC Interrupts and DMA Requests

Status Register (STAT)		Description	Can generate		
Flag	Name		Interrupt?	DMA Request?	Low Power Wakeup?
RDYn	Result FIFO n Ready Flag	Conversion result data is written to Result FIFO and has a watermark configurable trigger level to generate an exception request as controlled by FCTRL0[FWMARK].	Y	Y	Y
FOFn	Result FIFO n Overflow Flag	Attempts to store data to the FIFO when the FIFO is full is an error condition.	Y	N	Y
TCOMP_INT	Trigger Completion Flag	A trigger sequence has been completed (all associated commands have been run).	Y	N	Y
TEXC_INT	High Priority Trigger Flag	A high priority trigger exception has occurred.	Y	N	Y

34.4 External signals

The ADC module supports analog channel inputs. ADC requires supply and ground connections.

Table 216. ADC signal descriptions

Signal	Description	I/O
V_{DDA}	Analog Power Supply	I
V_{SSA}	Analog Ground	I
CH0A - CHnA[1]	A-side Analog Channel Inputs	I

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

1. where n is the maximum channel number supported in the chip. See the chip-specific information for the number of channels supported on your device.

34.4.1 Analog channel inputs (CHnA)

CMDLn[ADCH] and CMDLn[CTYPE] control selection of individual input channels. Each ADC command independently makes a channel and conversion type selection.

NOTE

Some input channels from on-chip sources, such as temperature sensors and reference voltage sources, may only be connected to individual instances of ADC. Some input channel options in the field-setting descriptions may not be available for your device. See chip-specific information for the channels supported on this device.

34.5 Initialization

34.5.1 Calibration

The ADC module has multiple calibration functions that must be executed as part of ADC setup to achieve the specified accuracy. Calibration must be run after any reset and before a conversion is initiated. Prior to offset calibration or calibration, the user must configure the clock source and frequency of ADC for the clock source availability and needs of the application. ADC must be enabled (CTRL[ADCEN] = 1h) before a calibration function runs. If calibration is requested while the ADC is actively converting, that sequence completes before starting calibration.

Averaging multiple conversions can achieve improved accuracy during the calibration routines. It is recommended to set CTRL[CAL_AVGS] for minimum of 256 averaging during calibration steps. If the application uses ADC in a wide variety of configurations, the configuration for which the highest accuracy is required should be selected. Alternatively, multiple calibrations can be done for the different configurations.

34.5.1.1 Offset calibration

The Offset Trim Register (OFSTRIM) register is used to trim for ADC comparator offset voltage. The ADC supports an offset calibration function in which the OFSTRIM register is automatically updated. Below are the steps for executing the function.

1. Configure for the desired averaging via CTRL[CAL_AVGS]. The minimum recommended setting is for 256 averaging (CTRL[CAL_AVGS] set to 0x8 or greater).
2. Initiate Offset Calibration by setting CTRL[CALOFS].
3. Poll the STAT[CAL_RDY] flag. When STAT[CAL_RDY] is asserted, the offset calibration function has completed and the OFSTRIM register has updated.

34.5.1.2 ADC Calibration

ADC includes hardware calibration logic in which the converter is calibrated for gain error and linearity error correction of the raw conversion result. The GCR0[GCALR] and CAL_GAR registers control calibration for the A-side converter. The ADC supports a calibration function in which the CAL_GAR registers are automatically updated. The calibration function also updates GCC0[GAIN_CAL] (associated with A-side calibration) . For A-side calibration, a software calculation is required to derive GCR0[GCALR] from GCC0[GAIN_CAL]. Below are the steps for completing calibration setup.

1. Execute Offset calibration steps. The Offset Trim Register (OFSTRIM) is used during calibration to trim for comparator offset voltage.
2. Configure for the desired averaging via CTRL[CAL_AVGS]. The minimum recommended setting is for 256 averaging (CTRL[CAL_AVGS] set to 0x8 or greater).
3. Initiate the calibration routine by writing 1 to CTRL[CAL_REQ]. CTRL[CAL_REQ] remains 1 until the CAL routine has been accepted by the ADC. After acceptance, CTRL[CAL_REQ] automatically becomes 0.
4. Poll GCC0[RDY] flag. When it is asserted, the hardware controlled A-side calibration operation is complete and CAL_GAR and GCC0[GAIN_CAL] registers are updated. The updated value in GCC0[GAIN_CAL] is needed for further software processing described in the following steps.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

5. Read GCC0[GAIN_CAL] and store for use in the gain_adjustment calculation.
6. Calculate the A-side gain_adjustment = (131072)/(131072-GCC0[GAIN_CAL]). GCC0[GAIN_CAL] is a 16-bit signed value. This results in a floating point value between 0 and 2.
7. Convert the floating point value to its integer component (0 or 1) and its fractional component rounded to 16-bits. The integer value is stored to GCR0[GCALR[16]] and the fractional component is stored to GCR0[GCALR[15:0]]. Write this value to the GCR0[GCALR] register.
8. Once GCR0[GCALR] contains the result from the gain_adjustment calculation, set the GCR0[RDY] flag to indicate it is valid.

After completing the steps above, the calibration sequence is complete and the STAT[CAL_RDY] flag is set. The STAT[CAL_RDY] flag remains set until the user resets the system or requests a new calibration sequence.

When STAT[CAL_RDY] is set, the ADC is configured to run in calibrated mode. Each conversion uses a combination of linearity and gain calibration results to correct SAR data. Calibration conversion latency is required to process each sample. However, due to the pipelined nature of data and control sequences, each conversion can still be initiated without experiencing this calibration delay.

34.5.1.3 Calibration General A-Side Register valid ranges

The general calibration value registers CAL_GARa have 34 registers in each array. The data in each register is a signed 16-bit value but has a variable range of possible values. The following table defines the valid range each register supports.

Table 217. Calibration General Registers Valid Values

Element CAL_GxR[N]	Valid Range
N = 0x00, 0x20	–1024 to +1023
N = 0x01	–2048 to +2047
N = 0x02–0x03	–4096 to +4095
N = 0x04–0x07	–8192 to +8191
N = 0x08–0x0F	–16384 to +16383
N = 0x10–0x1F, 0x21	–32768 to +32767

These registers are typically updated automatically during the self calibration sequence. To reduce the latency associated with ADC setup, the CAL_GxR values from a calibration sequence can be stored in non-volatile memory after an initial calibration and written to the CAL_GxR registers via software prior to the first ADC conversion. If these registers are set to values not generated by the calibration function, the linearity error specifications may not be met. Note, these values can only be written in a single access, byte accesses are not supported.

34.5.1.4 High speed calibration

The High Speed Trim Register (HSTRIM) register is used to trim for high speed mode conversions (when CFG2[HS] is set). The ADC supports a high speed calibration function in which the HSTRIM register is automatically updated. Below are the steps for executing the function.

1. Configure for the desired averaging via CTRL[CAL_AVGS]. The minimum recommended setting is for 256 averaging (CTRL[CAL_AVGS] set to 0x8 or greater).
2. Initiate High speed calibration by setting CTRL[CALHS].
3. Poll the STAT[CAL_RDY] flag. When STAT[CAL_RDY] is asserted, the high speed calibration function has completed and the HSTRIM register has updated.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

34.6 ADC register descriptions

This section describes the ADC registers.

NOTE

Writing to a read-only (RO) register can cause bus errors.

34.6.1 ADC memory map

HSADC0 base address: 400A_F000h

Offset	Register	Width (In bits)	Access	Reset value
0h	Version ID Register (VERID)	32	R	0200_1409h
4h	Parameter Register (PARAM)	32	R	0707_0804h
10h	Control Register (CTRL)	32	RW	0000_0000h
14h	Status Register (STAT)	32	RW	0000_0000h
18h	Interrupt Enable Register (IE)	32	RW	0000_0000h
1Ch	DMA Enable Register (DE)	32	RW	0000_0000h
20h	Configuration Register (CFG)	32	RW	0080_0000h
24h	Pause Register (PAUSE)	32	RW	0000_0000h
34h	Software Trigger Register (SWTRIG)	32	RW	0000_0000h
38h	Trigger Status Register (TSTAT)	32	RW	0000_0000h
40h	Offset Trim Register (OFSTRIM)	32	RW	0000_0000h
48h	High Speed Trim Register (HSTRIM)	32	RW	0000_0000h
A0h - ACh	Trigger Control Register (TCTRL0 - TCTRL3)	32	RW	0000_0000h
E0h	FIFO Control Register (FCTRL0)	32	RW	0000_0000h
F0h	Gain Calibration Control (GCC0)	32	R	0000_0000h
F8h	Gain Calculation Result (GCR0)	32	RW	0001_0000h
100h	Command Low Buffer Register (CMDL1)	32	RW	0000_0000h
104h	Command High Buffer Register (CMDH1)	32	RW	0000_0000h
108h	Command Low Buffer Register (CMDL2)	32	RW	0000_0000h
10Ch	Command High Buffer Register (CMDH2)	32	RW	0000_0000h
110h	Command Low Buffer Register (CMDL3)	32	RW	0000_0000h
114h	Command High Buffer Register (CMDH3)	32	RW	0000_0000h
118h	Command Low Buffer Register (CMDL4)	32	RW	0000_0000h
11Ch	Command High Buffer Register (CMDH4)	32	RW	0000_0000h
120h	Command Low Buffer Register (CMDL5)	32	RW	0000_0000h

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Offset	Register	Width (In bits)	Access	Reset value
124h	Command High Buffer Register (CMDH5)	32	RW	0000_0000h
128h	Command Low Buffer Register (CMDL6)	32	RW	0000_0000h
12Ch	Command High Buffer Register (CMDH6)	32	RW	0000_0000h
130h	Command Low Buffer Register (CMDL7)	32	RW	0000_0000h
134h	Command High Buffer Register (CMDH7)	32	RW	0000_0000h
200h - 218h	Compare Value Register (CV1 - CV7)	32	RW	0000_0000h
300h	Data Result FIFO Register (RESFIFO0)	32	R	0000_0000h
400h - 484h	Calibration General A-Side Registers (CAL_GAR0 - CAL_GAR33)	32	RW	0000_0000h
FF8h	Configuration 2 Register (CFG2)	32	RW	0000_1000h

34.6.2 Version ID Register (VERID)

Offset

Register	Offset
VERID	0h

Function

The Version ID register indicates the version integrated for this instance on the device and also indicates inclusion/exclusion of several optional features.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	MAJOR								MINOR							
W																
Reset	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	NUM_FIFO			NUM_SEC	CALOFSI	IADCKI	VR1RNGI	0	CSW			MVI	0	DIFFEN	RES
W																
Reset	0	0	0	1	0	1	0	0	0	0	0	0	1	0	0	1

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Fields

Field	Function
31-24 MAJOR	Major Version Number This read-only field returns the major version number for the module specification.
23-16 MINOR	Minor Version Number This read-only field returns the minor version number for the module specification.
15 —	Reserved This read-only field is reserved and always has the value 0.
14-12 NUM_FIFO	Number of FIFOs This read-only field indicates the number of result FIFOs implemented in the design. 000b - N/A 001b - This design supports one result FIFO. 010b - This design supports two result FIFOs. 011b - This design supports three result FIFOs. 100b - This design supports four result FIFOs.
11 NUM_SEC	Number of Single Ended Outputs Supported This read-only field indicates the number of single ended channels which can be processed simultaneously. 0b - This design supports one single ended conversion at a time. 1b - This design supports two simultaneous single ended conversions.
10 CALOFSI	Calibration Function Implemented This read-only field indicates if the ADC contains hardware calibration functions. When supported, CTRL[CALOFS] and CTRL[CAL_REQ] can be used to request the calibration routines to run. 0b - Calibration Not Implemented. 1b - Calibration Implemented.
9 IADCKI	Internal ADC Clock Implemented This read-only field indicates if this implementation of the ADC block includes an internal clock source. When supported, the CFG[ADCKEN] field is available and documented for enabling the clock source. When available, this clock source is used in clock selection logic external to the module for use within the system. 0b - Internal clock source not implemented. 1b - Internal clock source (and CFG[ADCKEN]) implemented.
8 VR1RNGI	Voltage Reference 1 Range Control Bit Implemented

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	This read-only field indicates if a control bit is implemented for selecting the input voltage range on Voltage Reference Option 1.When range control is required, the CFG[VREF1RNG] field is available and documented for controlling the Voltage Reference Option 1. 0b - Range control not required. CFG[VREF1RNG] is not implemented. 1b - Range control required. CFG[VREF1RNG] is implemented.
7 —	Reserved This read-only field is reserved and always has the value 0.
6-4 CSW	Channel Scale Width This read-only field indicates if channel scaling is supported by this implementation. When supported, each command buffer has a control field (CMDLa[CSCALE]) for setting input scaling. 000b - Channel scaling not supported. 001b - Channel scaling supported. 1-bit CSCALE control field. 110b - Channel scaling supported. 6-bit CSCALE control field.
3 MVI	Multi Vref Implemented This read-only field indicates if multiple Voltage Reference High inputs are supported by this implementation. When multiple voltage references are supported, the CFG[REFSEL] field is available and documented for selecting the Voltage Reference High options. 0b - Single voltage reference high (VREFH) input supported. 1b - Multiple voltage reference high (VREFH) inputs supported.
2 —	Reserved This read-only field is reserved and always has the value 0.
1 DIFFEN	Differential Supported This read-only field indicates if differential operation is supported by this implementation. When supported, CMDLa[CTYPE] in each command buffer is used for configuring for differential operation. 0b - Differential operation not supported. 1b - Differential operation supported.
0 RES	Resolution This read-only field indicates the maximum accuracy supported by this implementation. When RES=1, each command buffer has a control field (CMDLa[MODE]) for selecting resolution of conversions for the associated command. 0b - Up to 12-bit single ended resolution supported (and 13-bit differential resolution if VERID[DIFFEN] = 1b). 1b - Up to 16-bit single ended resolution supported (and 16-bit differential resolution if VERID[DIFFEN] = 1b).

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

34.6.3 Parameter Register (PARAM)

Offset

Register	Offset
PARAM	4h

Function

The Parameter register indicates the size of several variable integration options for this instance on the device.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	CMD_NUM								CV_NUM							
W																
Reset	0	0	0	0	0	1	1	1	0	0	0	0	0	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	FIFOSIZE								TRIG_NUM							
W																
Reset	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0

Fields

Field	Function
31-24 CMD_NUM	Command Buffer Number Number of command buffers implemented.
23-16 CV_NUM	Compare Value Number Number of compare value registers implemented.
15-8 FIFOSIZE	Result FIFO Depth The maximum number of conversion datawords that can be stored in the result FIFO before an overflow occurs. This field is read-only. 0000_0001b - Result FIFO depth = 2 dataword. 0000_0100b - Result FIFO depth = 4 datawords. 0000_1000b - Result FIFO depth = 8 datawords. 0001_0000b - Result FIFO depth = 16 datawords. 0010_0000b - Result FIFO depth = 32 datawords. 0100_0000b - Result FIFO depth = 64 datawords.
7-0 TRIG_NUM	Trigger Number Number of Triggers implemented.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

34.6.4 Control Register (CTRL)

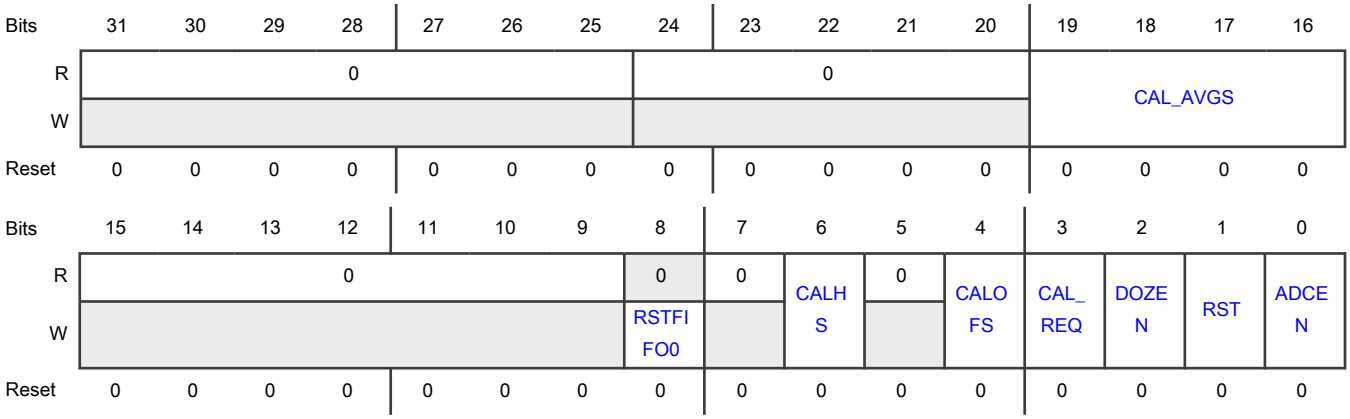
Offset

Register	Offset
CTRL	10h

Function

Control Register includes primary control bits.

Diagram



Fields

Field	Function
31-25 —	Reserved This read-only field is reserved and always has the value 0.
24-20 —	Reserved This read-only field is reserved and always has the value 0.
19-16 CAL_AVGS	Auto-Calibration Averages Selects how many ADC conversions are averaged to calculate each calibration value. Selecting a higher number of averages will lead to more accurate conversions after completing calibration. The recommended minimum setting is for 256 averaging (CAL_AVGS set to 0x8 or greater). The CAL_AVGS bit field applies to CALOFS, CAL_REQ and CALHS calibration functions. CAL_AVGS should be fixed when requesting and running calibration with CTRL[CAL_REQ], CTRL[CALOFS], or CTRL[CALHS]. . 0000b - Single conversion. 0001b - 2 conversions averaged. 0010b - 4 conversions averaged. 0011b - 8 conversions averaged.

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	0100b - 16 conversions averaged. 0101b - 32 conversions averaged. 0110b - 64 conversions averaged. 0111b - 128 conversions averaged. 1000b - 256 conversions averaged. 1001b - 512 conversions averaged. 1010b - 1024 conversions averaged.
15-9 —	Reserved This read-only field is reserved and always has the value 0.
8 RSTFIFO0	Reset FIFO 0 0b - No effect. 1b - FIFO 0 is reset.
7 —	Reserved This read-only field is reserved and always has the value 0.
6 CALHS	High Speed Mode Trim Request Setting CALHS to 1 initiates a high speed mode trim calculation. Any conversions in progress are completed before launching the high speed mode calibration function. After being accepted, the ADC calculates the trim value and updates the HSTRIM register automatically. The CALHS bit is cleared by hardware upon completion of the high speed mode trim calculation. The STAT[CAL_RDY] bit indicates when the high speed mode calibration routine has completed. 0b - No request for high speed mode trim has been made 1b - Request for high speed mode trim has been made
5 —	Reserved This read-only field is reserved and always has the value 0.
4 CALOFS	Offset Calibration Request Setting CALOFS to 1 initiates a hardware offset calibration calculation. Any conversions in progress are completed before launching the offset calibration function. After being accepted, the ADC calculates the offset value and updates either the OFSTRIM register automatically. The CALOFS bit is cleared by hardware upon completion of the offset calibration calculation. The STAT[CAL_RDY] bit indicates when the offset calibration routine has completed. 0b - No request for offset calibration has been made 1b - Request for offset calibration function
3 CAL_REQ	Auto-Calibration Request

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	Request the hardware calibration routine to run. This bit is automatically cleared when the system accepts the hardware calibration request. Note, the status flag STAT[CAL_RDY] will indicate when this calibration routine has been completed. 0b - No request for hardware calibration has been made 1b - A request for hardware calibration has been made
2 DOZEN	Doze Enable Controls system transition to low power modes while ADC is converting. When DOZEN is clear, immediate entries to low power modes are allowed. Note that the selected clock source provided from the on-chip clock source must be able to continue operating. When the DOZEN bit is set, the ADC waits for the current averaging iteration/FIFO storage to complete. When the system is entering a low power mode in which ADC operation is not supported, the DOZEN bit is ignored and the ADC waits for the current transfer to complete any pending operation. 0b - ADC is enabled in low power mode. 1b - ADC is disabled in low power mode.
1 RST	Software Reset Resets all internal logic and registers, except ADC_CTRL, ADC_GCRn and ADC_CAL_GARn. Remains set until cleared by software. 0b - ADC logic is not reset. 1b - ADC logic is reset.
0 ADCEN	ADC Enable 0b - ADC is disabled. 1b - ADC is enabled.

34.6.5 Status Register (STAT)

Offset

Register	Offset
STAT	14h

Function

The Status Register provides the current status of the ADC module.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0					CMDACT			0						TRGACT	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0				ADC_ ACT...	CAL_ RDY	TCOM P_I...	TEXC_ INT	0						FOF0	RDY0
W							W1C	W1C							W1C	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-27 —	Reserved This read-only field is reserved and always has the value 0.
26-24 CMDACT	Command Active CMDACT is a read-only status field indicating the command that is actively being processed. Note, commands are only shown here when they are actively being processed by the SAR routine. Use CMDACT in conjunction with ADC_ACTIVE to determine which conversion is running. 000b - No command is currently in progress. 001b - Command 1 currently being executed. 010b - Command 2 currently being executed. 011b-111b - Associated command number is currently being executed.
23-18 —	Reserved This read-only field is reserved and always has the value 0.
17-16 TRGACT	Trigger Active TRGACT is a read-only status field indicating the trigger actively being processed. This can be used to determine which trigger is running through a trigger delay or being converted by the SAR routine. The command associated with TRGACT will be running through a conversion when CMDACT is non-zero. 00b - Command (sequence) associated with Trigger 0 currently being executed. 01b - Command (sequence) associated with Trigger 1 currently being executed. 10b - Command (sequence) associated with Trigger 2 currently being executed. 11b - Command (sequence) associated with Trigger 3 currently being executed.
15-12 —	Reserved This read-only field is reserved and always has the value 0.
11	ADC Active

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
ADC_ACTIVE	This flag shows if the module is currently processing a conversion, or has pending triggers to service. When the ADC is IDLE, this bit is set to 0b0. 0b - The ADC is IDLE. There are no pending triggers to service and no active commands are being processed. 1b - The ADC is processing a conversion, running through the power up delay, or servicing a trigger.
10 CAL_RDY	Calibration Ready This flag shows if the ADC CTRL[CAL_REQ] or CTRL[CALOFS] request for calibration has been completed and the results are ready. This flag is automatically cleared when a new hardware calibration or OFSTRIM request is made. 0b - Calibration is incomplete or hasn't been ran. 1b - The ADC is calibrated.
9 TCOMP_INT	Interrupt Flag For Trigger Completion When a trigger sequence has been completed (all associated commands have been run) the TCOMP_INT flag is asserted. Note that IE[TCOMP_IE] must be set for the specific trigger source to flag an interrupt upon completion. 0b - Either IE[TCOMP_IE] is set to 0, or no trigger sequences have run to completion. 1b - Trigger sequence has been completed and all data is stored in the associated FIFO.
8 TEXC_INT	Interrupt Flag For High Priority Trigger Exception When a high priority trigger exception has occurred and CFG[TRES] = 0, this flag is asserted. This flag only asserts if trigger exception interrupts are enabled (IE[TEXC_IE] = 0x1). This flag can be used in conjunction with TSTAT[TEXC_NUM] to resolve which trigger source was interrupted. 0b - No trigger exceptions have occurred. 1b - A trigger exception has occurred and is pending acknowledgement.
7-2 —	Reserved This read-only field is reserved and always has the value 0.
1 FOF0	Result FIFO 0 Overflow Flag Indicates that more data has been written to the Result FIFO 0 than it can hold. The newer data is not stored and the FIFO remains holding the original contents. This flag asserts regardless of the value of IE[FOFIE0]. However, an interrupt request is issued only if IE[FOFIE0] is set. This flag is cleared by writing a 1. 0b - No result FIFO 0 overflow has occurred since the last time the flag was cleared. 1b - At least one result FIFO 0 overflow has occurred since the last time the flag was cleared.
0 RDY0	Result FIFO 0 Ready Flag Indicates when the number of valid datawords in the result FIFO 0 is greater than the watermark level set in the FCTRL[FWMARK] bitfield. This flag asserts regardless of the value of IE[FWMIE0]. However,

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	an interrupt request or DMA request occurs only when the associated control bit (IE[FWMIE0] and DE[FWMDE0]) is set. This flag is cleared when the FCTRL0[FCOUNT] (which decrements on each read of the RESFIFO register) is less than or equal to the watermark level set in the FCTRL[FWMARK] bitfield. 0b - Result FIFO 0 data level not above watermark level. 1b - Result FIFO 0 holding data above watermark level.

34.6.6 Interrupt Enable Register (IE)

Offset

Register	Offset
IE	18h

Function

Interrupt Enable Register includes system interrupt masking control bits.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0												TCOMP_IE			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0							TEXC_IE	0						FOFIE0	FWMIE0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-20 —	Reserved This read-only field is reserved and always has the value 0.
19-16 TCOMP_IE	Trigger Completion Interrupt Enable Each bit in TCOMP_IE corresponds to a trigger source. (TCOMP_IE[0] corresponds to trigger 0, TCOMP_IE[1] corresponds to trigger 1, ...) A trigger completion interrupt is used to indicate when a complete trigger command sequence has been executed. All results will be stored in the FIFO when a sequence is considered complete.

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	0000b - Trigger completion interrupts are disabled. 0001b - Trigger completion interrupts are enabled for trigger source 0 only. 0010b - Trigger completion interrupts are enabled for trigger source 1 only. 0011b-1110b - Associated trigger completion interrupts are enabled. 1111b - Trigger completion interrupts are enabled for every trigger source.
15-9 —	Reserved This read-only field is reserved and always has the value 0.
8 TEXC_IE	Trigger Exception Interrupt Enable This register field enables the ADC to assert an interrupt request when a high priority trigger exception occurs. TSTAT[TEXC_NUM] will contain the value of the corresponding trigger which was affected by the exception. 0b - Trigger exception interrupts are disabled. 1b - Trigger exception interrupts are enabled.
7-2 —	Reserved This read-only field is reserved and always has the value 0.
1 FOFIE0	Result FIFO 0 Overflow Interrupt Enable Configures the module to generate overflow interrupt requests when FOF flag is asserted. 0b - FIFO 0 overflow interrupts are not enabled. 1b - FIFO 0 overflow interrupts are enabled.
0 FWMIE0	FIFO 0 Watermark Interrupt Enable Configures the module to generate watermark interrupt requests when STAT[RDY0] flag is asserted. 0b - FIFO 0 watermark interrupts are not enabled. 1b - FIFO 0 watermark interrupts are enabled.

34.6.7 DMA Enable Register (DE)

Offset

Register	Offset
DE	1Ch

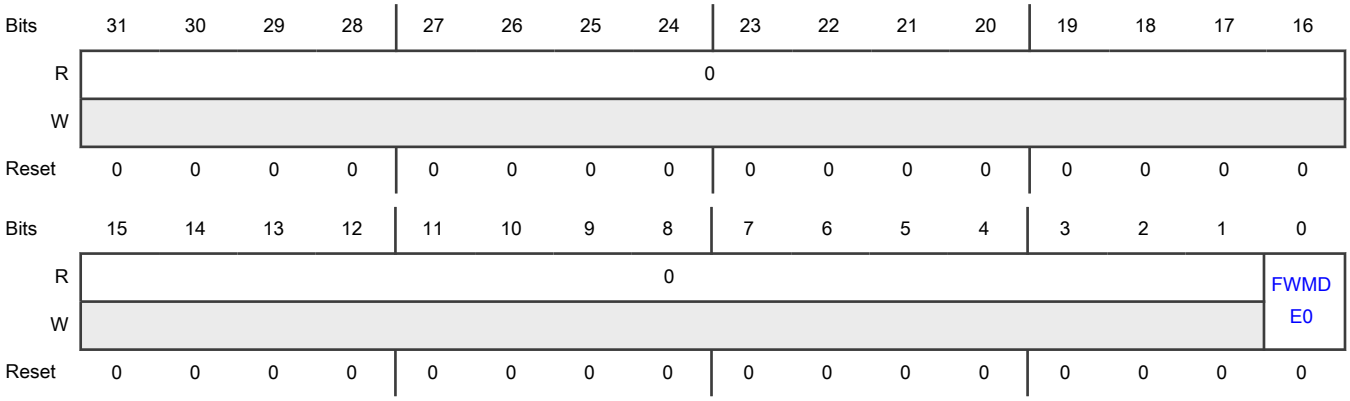
Function

DMA Enable Register includes DMA request masking control bits.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram



Fields

Field	Function
31-1 —	Reserved This read-only field is reserved and always has the value 0.
0 FWMDE0	FIFO 0 Watermark DMA Enable Configures the module to generate DMA requests when STAT[RDY0] flag is asserted. 0b - DMA request disabled. 1b - DMA request enabled.

34.6.8 Configuration Register (CFG)

Offset

Register	Offset
CFG	20h

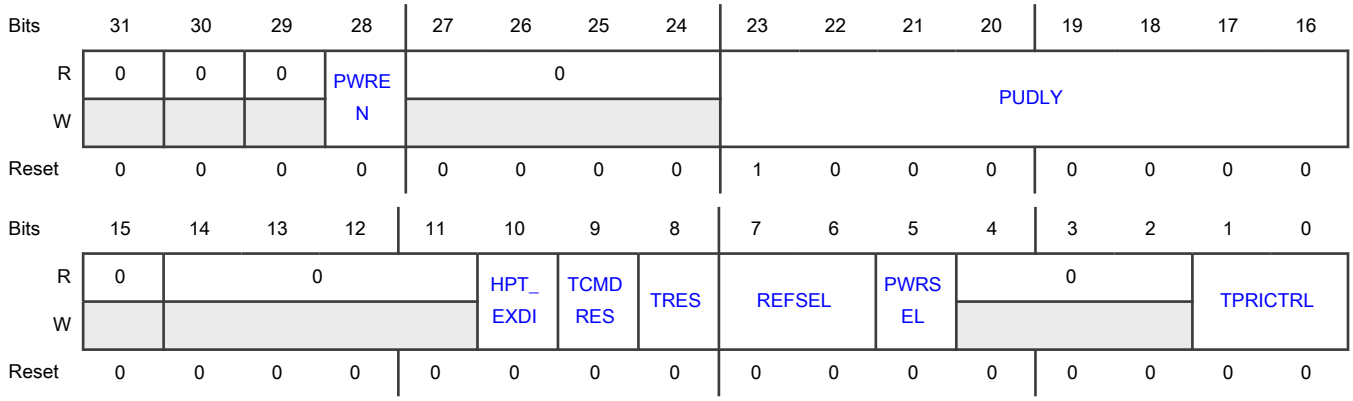
Function

The Configuration Register controls ADC functions that are common to all commands. The CFG cannot be changed while the CTRL[ADCEN] bit is set. Writes to CFG while CTRL[ADCEN] is set are ignored.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram



Fields

Field	Function
31 —	Reserved This read-only field is reserved and always has the value 0.
30 —	Reserved This read-only field is reserved and always has the value 0.
29 —	Reserved This read-only field is reserved and always has the value 0.
28 PWREN	ADC Analog Pre-Enable Enables the ADC analog circuits. When setting CFG[PWREN], user code should delay for a period exceeding the analog startup time of $t_{ADCSTUP}$ before enabling the ADC for operation. The module is still operational even when CFG[PWREN] is left clear but command execution start is delayed for a period defined by CFG[PUDLY]. Refer to the device datasheet for ADC idle and ADC active power consumption parameters. 0b - ADC analog circuits are only enabled while conversions are active. Performance is affected due to analog startup delays. 1b - ADC analog circuits are pre-enabled and ready to execute conversions without startup delays (at the cost of higher DC current consumption). Note that a single power up delay (CFG[PUDLY]) is executed immediately once PWREN is set, and any detected trigger does not begin ADC operation until the power up delay time has passed. After this initial delay expires the analog remains pre-enabled and no additional delays are executed.
27-24 —	Reserved This read-only field is reserved and always has the value 0.
23-16 PUDLY	Power Up Delay The PUDLY must be programmed to a non-zero value to enable the ADC. Note, the PUDLY is executed in one of two modes depending on the value configured in PWREN:

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	When CFG[PWREN]=0b0, the ADC analog circuits are only powered on while the module is active. The delay defined by CFG[PUDLY] is executed after an initial trigger transitions the ADC from its Idle state. This delay allows time for the analog circuits to stabilize after being powered on. The startup delay count of (PUDLY*4) ADCK cycles must result in a longer delay than the analog startup time of $t_{ADCSTUP}$. Accuracy of the initial conversion(s) after activation is degraded if CFG[PUDLY] is set to too small a value. At the end of active conversions, if there are no subsequent pending conversions, the ADC analog is automatically reverted back to its low power idle state and the PUDLY runs again after the ADC is awakened with a later trigger. When CFG[PWREN]=0b1 prior to ADC activation, the analog circuits are pre-enabled and the activation delay defined by CFG[PUDLY] is executed immediately. After the delay has been executed, the analog remains enabled regardless of ADC activity. This configuration has the benefit of beginning conversions immediately after trigger event detection at the cost of increased DC power consumption of the analog circuits. Note, the ADC does not begin an initial conversion until the PUDLY has been executed regardless of the PWREN setting.
15 —	Reserved This read-only field is reserved and always has the value 0.
14-11 —	Reserved This read-only field is reserved and always has the value 0.
10 HPT_EXDI	High Priority Trigger Exception Disable Determines if high priority trigger exceptions are disabled. See section Trigger detect and command execution for a detailed description on trigger exception handling. If HPT_EXDI = 1, then exceptions are disabled and fields CFG[TCMDRES], CFG[TRES], and CFG[TPRICTRL] are ignored. 0b - High priority trigger exceptions are enabled. 1b - High priority trigger exceptions are disabled.
9 TCMDRES	Trigger Command Resume Determines where to resume from a high priority trigger exception. CFG[TRES] must be asserted for TCMDRES to be used. If asserted, the interrupted trigger sequence is resumed from the command where it was interrupted. If de-asserted the trigger sequence is restarted from the beginning. 0b - Trigger sequences interrupted by a high priority trigger exception is automatically restarted. 1b - Trigger sequences interrupted by a high priority trigger exception is resumed from the command executing before the exception.
8 TRES	Trigger Resume Enable Determines what happens after completing a high priority trigger exception. If CFG[TRES] is asserted, the interrupted trigger sequence is resumed or restarted automatically. The sequence can be resumed along command boundaries or restarted from the beginning of a sequence. This is controlled by CFG[TCMDRES]. If CFG[TRES] is de-asserted an interrupted trigger sequence does not resume automatically. In this case, interrupted triggers can be resumed via software by monitoring STAT[TEXC_INT] bit (or ISR handling for a trigger exception interrupt when IE[TEXC_IE] is set). Software determines which trigger was

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	interrupted by reading TSTAT[TEXC_NUM] and re-starts the trigger with a write of 1 to the corresponding SWTRIG[SWTx] bit. 0b - Trigger sequences interrupted by a high priority trigger exception are not automatically resumed or restarted. 1b - Trigger sequences interrupted by a high priority trigger exception are automatically resumed or restarted.
7-6 REFSEL	Voltage Reference Selection Selects the voltage reference high used for conversions. **NOTE** See the chip-specific ADC information on the voltage reference options specific to this packaged device. 00b - (Default) Option 1 setting. 01b - Option 2 setting. 10b - Option 3 setting. 11b - Reserved
5 PWRSEL	Power Configuration Select Configures the module for power and performance. In the high-power setting, the highest conversion rates are possible. See the device data sheet for power and performance capabilities for each setting. 0b - Low power 1b - High power
4-2 —	Reserved
1-0 TPRICTRL	ADC Trigger Priority Control Controls how higher priority trigger exceptions are handled. See Trigger detect and command execution for a detailed explanation trigger event handling. 00b - If a higher priority trigger is detected during command processing, the current conversion is aborted and the new command specified by the trigger is started. 01b - If a higher priority trigger is received during command processing, the current command is stopped after completing the current conversion. If averaging is enabled, the averaging loop will be completed. However, CMDHa[LOOP] will be ignored and the higher priority trigger will be serviced. 10b - If a higher priority trigger is received during command processing, the current command will be completed (averaging, looping, compare) before servicing the higher priority trigger. 11b - Reserved

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

34.6.9 Pause Register (PAUSE)

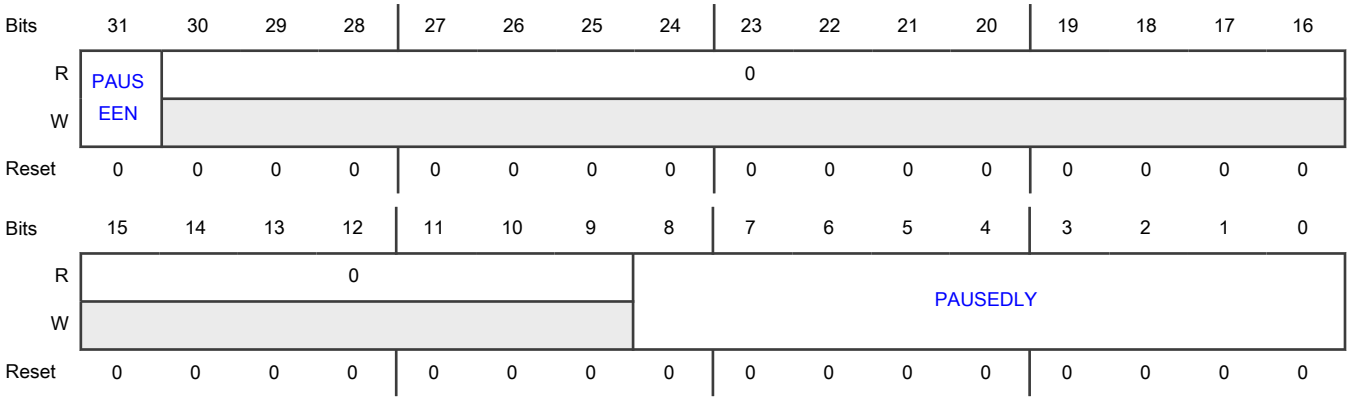
Offset

Register	Offset
PAUSE	24h

Function

The Pause Register controls an optional inserted delay between conversions. Note, the PAUSE register should not be modified while the CTRL[ADCEN] bit is set.

Diagram



Fields

Field	Function
31 PAUSEEN	PAUSE Option Enable Enables the ADC pausing function. When enabled, a programmable delay is inserted during command execution sequencing between LOOP iterations, between commands in a sequence, and between conversions when command is executing in for "Compare Until True" configuration. Note, CMDHa[WAIT_TRIG] and PAUSE are mutually exclusive. CMDHa[WAIT_TRIG] will take priority over PAUSE between commands when both are enabled. 0b - Pause operation disabled 1b - Pause operation enabled
30-9 —	Reserved This read-only field is reserved and always has the value 0.
8-0 PAUSEDLY	Pause Delay When PAUSEEN is set, the PAUSEDLY field controls the duration of pausing during command execution sequencing. The pause delay is a count of (PAUSEDLY*4) ADCK cycles.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

34.6.10 Software Trigger Register (SWTRIG)

Offset

Register	Offset
SWTRIG	34h

Function

The Software Trigger Register (SWTRIG) is written to initiate software triggered conversions. Writes to SWTRIG register are ignored while CTRL[ADCEN] is clear. Note that there is an approximately 3 ADC Clock cycle synchronization delay between asserting CTRL[ADCEN] until SWTRIG can be accepted.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0												0	0	0	0
W													SWT3	SWT2	SWT1	SWT0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-4 —	Reserved This read-only field is reserved and always has the value 0.
3 SWT3	Software Trigger 3 Event Write 1 to SWT3 generates a trigger 3 event. Writing 1 to SWT3 is ignored while the trigger 3 event is being serviced or is pending. 0b - No trigger 3 event generated. 1b - Trigger 3 event generated.
2 SWT2	Software Trigger 2 Event Write 1 to SWT2 generates a trigger 2 event. Writing 1 to SWT2 is ignored while the trigger 2 event is being serviced or is pending. 0b - No trigger 2 event generated. 1b - Trigger 2 event generated.
1	Software Trigger 1 Event

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
SWT1	Write 1 to SWT1 generates a trigger 1 event. Writing 1 to SWT1 is ignored while the trigger 1 event is being serviced or is pending. 0b - No trigger 1 event generated. 1b - Trigger 1 event generated.
0 SWT0	Software Trigger 0 Event Write 1 to SWT0 generates a trigger 0 event. Writing 1 to SWT0 is ignored while the trigger 0 event is being serviced or is pending. 0b - No trigger 0 event generated. 1b - Trigger 0 event generated.

34.6.11 Trigger Status Register (TSTAT)

Offset

Register	Offset
TSTAT	38h

Function

This register contains status flags to indicate when trigger sequences have been completed or interrupted by a high priority trigger exception. Each bit in this register is set by hardware and cleared by software. To clear a bit in this register, write a 0b1 to the corresponding bit position.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0												TCOMP_FLAG			
W													W1C			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0												TEXC_NUM			
W													W1C			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-20	Reserved

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
—	This read-only field is reserved and always has the value 0.
19-16 TCOMP_FLAG	Trigger Completion Flag Each bit corresponds to a trigger. When the corresponding trigger sequence has been completed a bit in TCOMP_FLAG will be set. For example, if trigger 3 has been completed, then bit 19 will be set in TSTAT. This register will only be active if the corresponding bit in IE[TCOMP_IE] = 1. Note, synchronization delay may be added to the end of the final conversion in a sequence prior to this flag being set. This delay can range from 2-4 ADC CLK cycles. 0000b - No triggers have been completed. Trigger completion interrupts are disabled. 0001b - Trigger 0 has been completed and trigger 0 has enabled completion interrupts. 0010b - Trigger 1 has been completed and trigger 1 has enabled completion interrupts. 0011b-1110b - Associated trigger sequence has completed and has enabled completion interrupts. 1111b - Every trigger sequence has been completed and every trigger has enabled completion interrupts.
15-4 —	Reserved This read-only field is reserved and always has the value 0.
3-0 TEXC_NUM	Trigger Exception Number Each bit corresponds to a trigger. When the corresponding trigger sequence has been interrupted by a high priority trigger exception a 1 will be set in its bit position. For example, if trigger 3 has been interrupted, then bit 3 will be set in TSTAT. This register will be active regardless of the value in IE[TEXC_IE]. 0000b - No triggers have been interrupted by a high priority exception. Or CFG[TRES] = 1. 0001b - Trigger 0 has been interrupted by a high priority exception. 0010b - Trigger 1 has been interrupted by a high priority exception. 0011b-1110b - Associated trigger sequence has interrupted by a high priority exception. 1111b - Every trigger sequence has been interrupted by a high priority exception.

34.6.12 Offset Trim Register (OFSTRIM)

Offset

Register	Offset
OFSTRIM	40h

Function

The Offset Trim register is used to trim for comparator offset. The ADC supports an offset calibration function in which the OFSTRIM register is automatically updated.

All information provided in this document is subject to legal disclaimers.

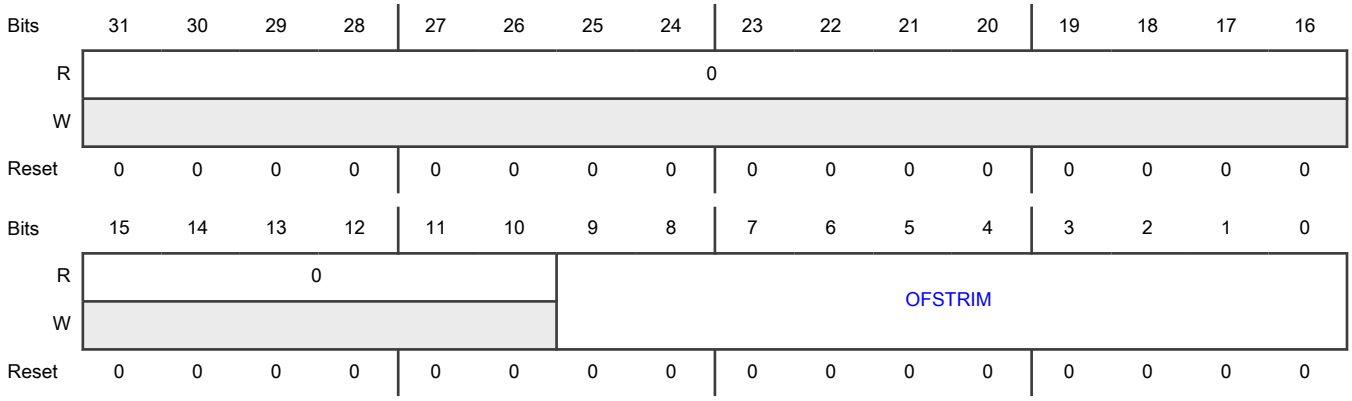
© 2026 NXP B.V. All rights reserved.

To trigger the automatic update of OFSTRIM, set the CTRL[CALOFS] bit. Upon completion of the offset calibration sequence, the STAT[CAL_RDY] bit is set to 0b1 and the OFSTRIM[OFSTRIM] bitfield is updated with a 10-bit signed value.

The OFSTRIM[OFSTRIM] value is used to minimize offset for normal conversions with a range of -512 to 511. For 16-bit conversions, each increment is 1/2 LSB resulting in a programmable offset range of -256 to +255.5 LSB. For 12-bit conversions, each increment is 1/32 LSB resulting in a programmable offset range of -16 to +15.96875 LSB.

The OFSTRIM[OFSTRIM] trim value is also used in the calibration function.

Diagram



Fields

Field	Function
31-10 —	Reserved This read-only field is reserved and always has the value 0.
9-0 OFSTRIM	Trim for Offset OFSTRIM is a 10-bit signed value between -512 and 511. This value is subtracted from the raw ADC conversion result.

34.6.13 High Speed Trim Register (HSTRIM)

Offset

Register	Offset
HSTRIM	48h

Function

The High Speed Trim register is used to trim for high speed function. The ADC supports an additional offset calibration function for high speed mode in which the HSTRIM register is automatically updated.

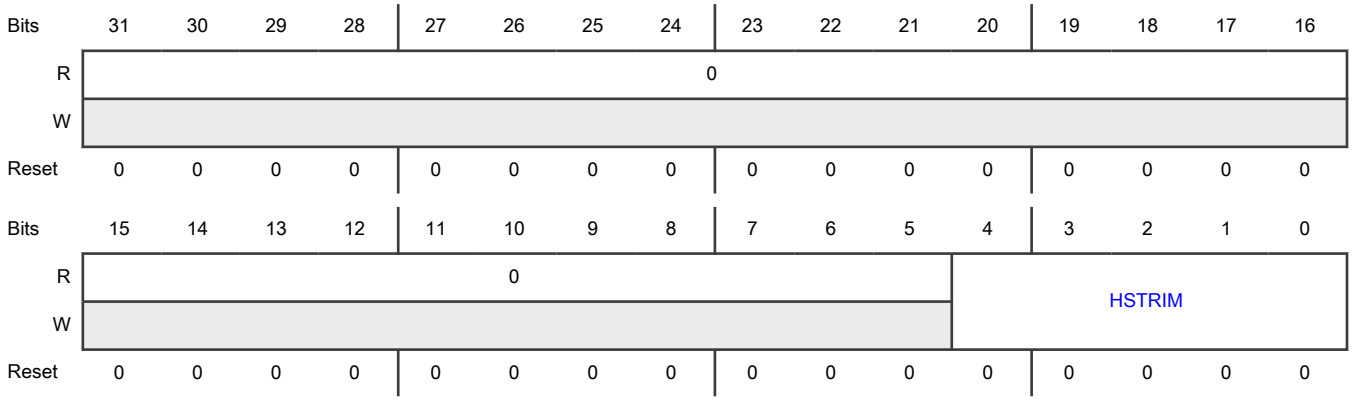
To trigger the automatic update of HSTRIM, set the CTRL[CALHS] bit. Upon completion of the high speed calibration sequence, the STAT[CAL_RDY] bit is set to 0b1 and the HSTRIM[HSTRIM] bitfield is updated with a 5-bit signed value.

The HSTRIM[HSTRIM] value is used to minimize offset for high speed conversions (when CFG2[HS] is set). The HSTRIM register supports a range of -16 to 15.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram



Fields

Field	Function
31-5 —	Reserved This read-only field is reserved and always has the value 0.
4-0 HSTRIM	Trim for High Speed Conversions HSTRIM is a 5-bit signed value between -16 and 15 that trims circuitry for high speed mode.

34.6.14 Trigger Control Register (TCTRL0 - TCTRL3)

Offset

Register	Offset
TCTRL0	A0h
TCTRL1	A4h
TCTRL2	A8h
TCTRL3	ACh

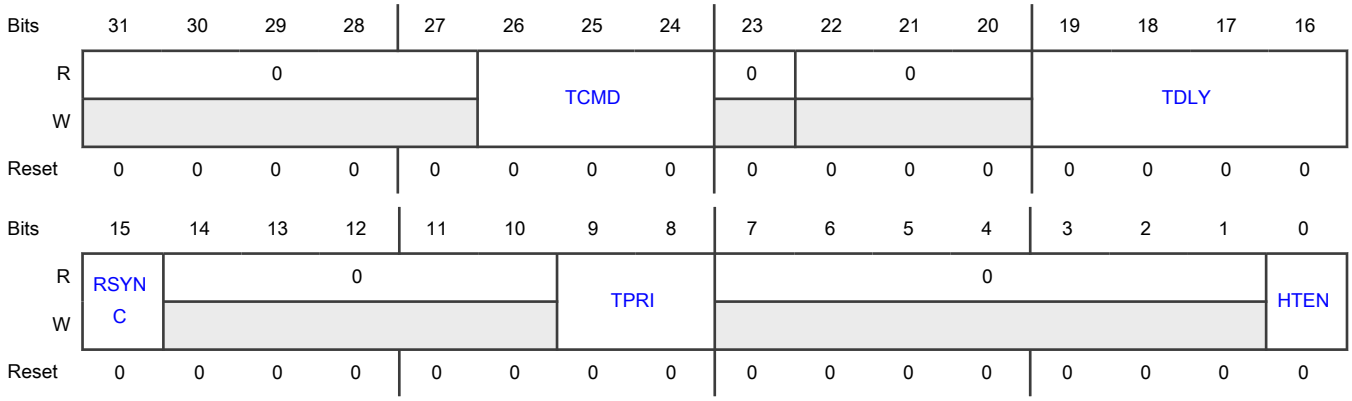
Function

The Trigger Control (TCTRL*a*) register implements control fields associated with each implemented trigger source. When the ADC is actively executing commands, only one of the TCTRL*a* registers is actively controlling ADC conversions. The actively controlling TCTRL*a* register should not be updated while the ADC is active. A write to a TCTRL*a* registers while that trigger control register is controlling the ADC operation may cause unpredictable behavior.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram



Fields

Field	Function
31-27 —	Reserved This read-only field is reserved and always has the value 0.
26-24 TCMD	Trigger Command Select Selects the command from command buffer to execute upon detection of the associated trigger event. When a trigger is received while the ADC is busy converting and the new trigger has higher priority than the trigger associated with the current command being executed, the command in this register, associated with the new trigger, is executed under control of CFG[TPRICTRL]. See section Trigger detect and command execution for a more detailed description on the relationship between CFG[TPRICTRL] and TCMD. 000b - Not a valid selection from the command buffer. Trigger event is ignored. 001b - CMD1 is executed 010b-110b - Corresponding CMD is executed 111b - CMD7 is executed
23 —	Reserved This read-only field is reserved and always has the value 0.
22-20 —	Reserved This read-only field is reserved and always has the value 0.
19-16 TDLY	Trigger Delay Select Selects the trigger delay duration to wait at the start of servicing a trigger event. Each trigger source has an associated programmable delay prior to beginning an initial conversion. When TDLY field is clear, then no delay is incurred. When TDLY is set to a non-zero value, the duration for the delay is 2^{TDLY} ADCK cycles.
15 RSYNC	Trigger Resync

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	If RSYNC is set to 1, this trigger source will be used as a resync trigger. A resync trigger can be configured to restart or abort a running trigger sequence (the target). Each resync trigger is configured to have a resync target. When a resync trigger is asserted, the target sequence will be aborted, the target FIFO destination(s) will be cleared, and the target sequence can optionally be restarted depending on CFG[TRES]. Multiple conditions must be met for a resync trigger to execute properly. 1. The resync trigger must have higher priority than the resync target. 2. The resync trigger TCTRLa[RSYNC] must be set to 0b1. 3. The resync target, specified by TCTRLa[TCMD], must be executing when the resync trigger is asserted. Please see Resync functionality for more information.
14-10 —	Reserved This read-only field is reserved and always has the value 0.
9-8 TPRI	Trigger Priority Setting This bitfield sets the priority of the associated trigger source. If two or more triggers have the same priority level setting, the lower order trigger event has the higher priority (e. g., if Trigger 0 and Trigger 1 are both pending triggers and TCTRL0[TPRI] is configured the same as TCTRL1[TPRI], then the command associated with Trigger 0 is serviced first). 00b - Set to highest priority, Level 1 01b-10b - Set to corresponding priority level 11b - Set to lowest priority, Level 4
7-1 —	Reserved This read-only field is reserved and always has the value 0.
0 HTEN	Trigger Enable Enables hardware trigger source to initiate conversion on the rising-edge of the input trigger source. **NOTE** Enabling hardware trigger will not disable software triggers. 0b - Hardware trigger source disabled 1b - Hardware trigger source enabled

34.6.15 FIFO Control Register (FCTRL0)

Offset

Register	Offset
FCTRL0	E0h

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Function

The FIFO Control (FCTRL*a*) registers contain control and status fields for each FIFO in the design. A programmable watermark can be set for each FIFO which can be used to trigger an interrupt. In addition, the number of entries stored in each FIFO can be monitored by reading FCTRL0[FCOUNT].

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0													FWMARK		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0												FCOUNT			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-19 —	Reserved This read-only field is reserved and always has the value 0.
18-16 FWMARK	Watermark Level Selection FWMARK is a programmable threshold setting. When the number of datawords stored in the ADC Result FIFO is greater than the value in FWMARK, the STAT[RDY] flag is asserted to indicate stored data has reached the programmable threshold. When IE[FWMIEx] is set, an interrupt request is generated. When DE[FWMDEx] is set, a DMA request is generated.
15-4 —	Reserved This read-only field is reserved and always has the value 0.
3-0 FCOUNT	Result FIFO Counter This read-only field indicates the number of datawords that are stored in the result FIFO. This value may be used in conjunction with PARAM[FIFOSIZE] to calculate how much room is left in the result FIFO. FCOUNT is incremented with each store of new data to the result FIFO and decrements with each read of the result FIFO. The FIFO is reset by writing to the CTRL[RSTFIFOx] bit, resulting in FCTRL0[FCOUNT] initialized to 0x0.

34.6.16 Gain Calibration Control (GCC0)

Offset

Register	Offset
GCC0	F0h

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Function

The Gain Calibration Control register (GCC0) hold intermediate values that are utilized as part of the calibration steps. The GCC*a*[GAIN_CAL] field is calculated with hardware and auto updated during the calibration steps. Once the hardware calibration sequence has updated GCC*a*[GAIN_CAL] the GCC*a*[RDY] status flag is asserted automatically. Note that requesting a calibration routine automatically clears the GCC*a*[RDY] bit until the new GCC*a*[GAIN_CAL] value is calculated. For the calibration steps to be completed, further software processing is needed where the GCC*a*[GAIN_CAL] value is used to calculate the gain adjustment and the result is stored to GCR*a*[GCALR]. For more information see Calibration in the functional description.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0							RDY	0							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	GAIN_CAL															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-25 —	Reserved This read-only field is reserved and always has the value 0.
24 RDY	Gain Calibration Value Valid This status flag indicates when the data stored in GCCa[GAIN_CAL] is valid. 0b - The GAIN_CAL value is invalid. Run the hardware calibration routine for this value to be set. 1b - The GAIN_CAL value is valid. GAIN_CAL should be used by software to derive GCRa[GCALR].
23-16 —	Reserved This read-only field is reserved and always has the value 0.
15-0 GAIN_CAL	Gain Calibration Value GAIN_CAL is read-only. As part of the hardware calibration steps, GAIN_CAL is automatically updated with a 16-bit signed number and is used in the gain adjustment calculations to derive the value written to the GCRa[GCALR] register. For more information see Calibration in the functional description.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

34.6.17 Gain Calculation Result (GCR0)

Offset

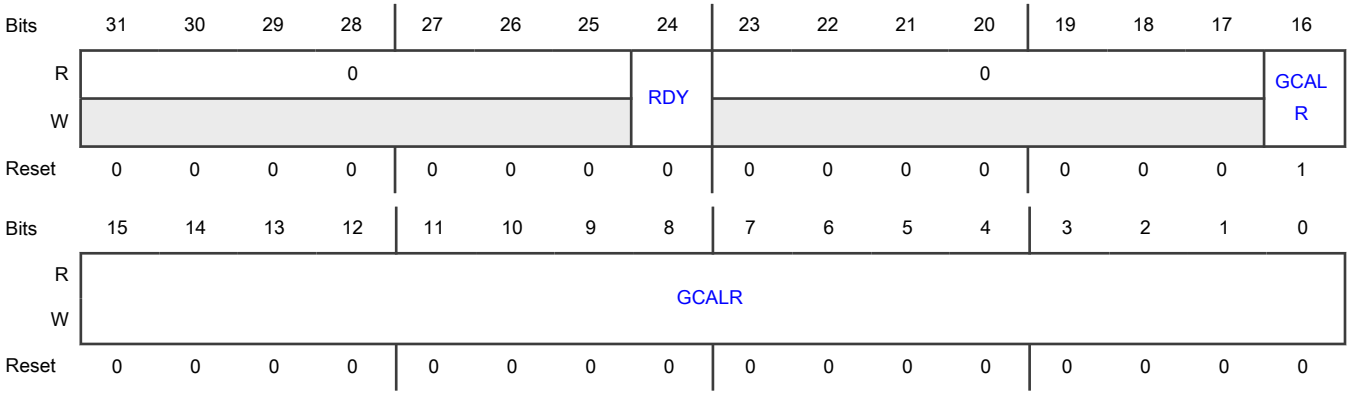
Register	Offset
GCR0	F8h

Function

The Gain Calculation Result register (GCR0) is used during ADC conversions for automated gain error adjustment. Determining the values to store to GCC0[GCALR] is the result of software calculations described in setup steps in Calibration. There is a RDY status flag in the GCR register which indicates whether the value GCRa[GCALR] is valid. Upon writing the GCRa[GCALR] value, the user should also assert GCRa[RDY] to indicate that the gain adjustment result is valid.

After beginning a calibration sequence by asserting CTRL[CAL_REQ], the calibration sequence is not complete until GCRa[GCALR] is calculated and GCRa[RDY] is asserted. The gain adjustment calculation results in a floating point value between 0 and 2. GCRa[GCALR][16] holds the integer value (0 or 1) and GCRa[GCALR][15:0] holds the 16-bit fractional component of the gain adjustment calculation. To convert the GCRa[GCALR] value back into the gain adjustment value in decimal format, this would be calculated as: 1*GCRa[16]+0.5*GCRa[15]+0.25*GCRa[14]+0.125*GCRa[13]+...

Diagram



Fields

Field	Function
31-25 —	Reserved This read-only field is reserved and always has the value 0.
24 RDY	Gain Calculation Ready The RDY flag indicates to the ADC when the data stored in GCRa[GCALR] is valid and is used for gain adjustment. The GCRa[GCALR] must be written from memory or calculated each time the calibration routine is run. The RDY flag must be asserted by the user after writing valid data into the GCALR field. The RDY flag is cleared automatically when requesting a new calibration sequence. 0b - The GCALR value is invalid. 1b - The GCALR value is valid.

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
23-17 —	Reserved This read-only field is reserved and always has the value 0.
16-0 GCALR	Gain Calculation Result This value holds the 17-bit value generated from the gain adjustment calculation during the calibration steps.

34.6.18 Command Low Buffer Register (CMDL1 - CMDL7)

Offset

Register	Offset
CMDL1	100h
CMDL2	108h
CMDL3	110h
CMDL4	118h
CMDL5	120h
CMDL6	128h
CMDL7	130h

Function

There are 7 command buffers (CMD*a*), each constructed from two 32-bit registers (CMDL*a*:CMDH*a*) that can be configured for different channel select and varying conversion options. Any of the command buffers is selected and used as the controlling command by association to a trigger event via configuration of the TCTRL*a*[TCMD] bitfield. When the ADC is actively executing commands, only one of the CMD buffers is actively controlling ADC conversions. The actively controlling CMD buffer should not be updated while the ADC is active. A write to a CMD buffer while that CMD buffer is controlling the ADC operation may cause unpredictable behavior.

NOTE

The 7 command buffers are numbered [CMDH1:CMDL1] through [CMDH7:CMDL7]. In NXP-supplied header files, these are likely to be defined as two 7-element arrays that are indexed from 0. I.e., the type declaration would be:

```
unsigned int CMDL[7];
unsigned int CMDH[7];
```

and software would access ADC0 CMDH1 as `ADC0->CMDH[0]` and ADC0 CMDL7 as `ADC0->CMDL[6]`.

The CMDL*a*[ADCH] and CMDL*a*[CTYPE] bitfields control selection of input channels. Each ADC command independently makes a channel selection.

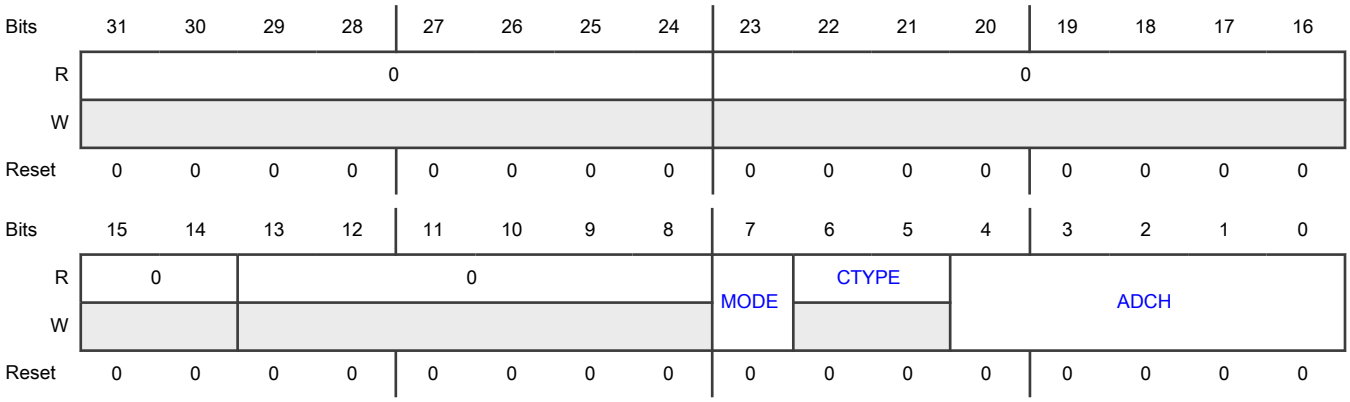
All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

NOTE

Some of the input channels are from on-chip sources such as temperature sensors and reference voltage sources and may only be connected to individual instances of the ADC module. Some of the input channel options in the bitfield-setting descriptions may not be available for your device. Refer to chip-specific information for the channels supported on this device.

Diagram



Fields

Field	Function
31-24 —	Reserved This read-only field is reserved and always has the value 0.
23-16 —	Reserved This read-only field is reserved and always has the value 0.
15-14 —	Reserved This read-only field is reserved and always has the value 0.
13-8 —	Reserved This read-only field is reserved and always has the value 0.
7 MODE	Select Resolution of Conversions The MODE field, selects the ADC resolution. 0b - Standard resolution. Single-ended 12-bit conversion. 1b - High resolution. Single-ended 16-bit conversion.
6-5 CTYPE	Conversion Type Zero is the only valid program value and selects A-side channels for conversion. 00b - Single-Ended Mode. Only A side channel is converted. 01b-11b - Reserved.
4-0	Input Channel Select

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
ADCH	**NOTE** Some of the input channels are from internal resources such as temperature sensors and bandgap voltage sources and may only be connected to individual instances of the ADC module. Some of the input channel options in the bitfield-setting descriptions might not be available for your device. For the ADC channel assignments for your device, see the chip-specific information. 0_0000b - Select CH0A. 0_0001b - Select CH1A. 0_0010b - Select CH2A. 0_0011b - Select CH3A. 0_0100b-1_1101b - Select corresponding channel CHnA. 1_1110b - Select CH30A. 1_1111b - Select CH31A.

34.6.19 Command High Buffer Register (CMDH1 - CMDH7)

Offset

Register	Offset
CMDH1	104h
CMDH2	10Ch
CMDH3	114h
CMDH4	11Ch
CMDH5	124h
CMDH6	12Ch
CMDH7	134h

Function

There are 7 command buffers (CMD*a*), each constructed from two 32-bit registers (CMDL*a*:CMDH*a*) that can be configured for different channel select and varying conversion options. Any of the command buffers is selected and used as the controlling command by association to a trigger event via configuration of the TCTRL*a*[TCMD] bitfield. When the ADC is actively executing commands, only one of the CMD buffers is actively controlling ADC conversions. The actively controlling CMD buffer should not be updated while the ADC is active. A write to a CMD buffer while that CMD buffer that is controlling ADC operation may result in unpredictable behavior.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

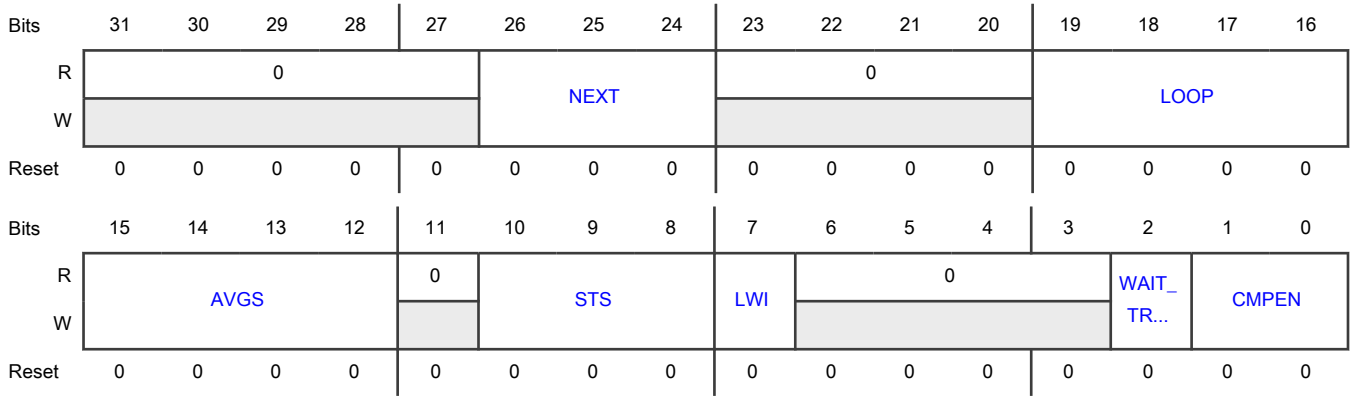
NOTE

The 7 command buffers are numbered [CMDH1:CMDL1] through [CMDH7:CMDL7]. In NXP-supplied header files, these are likely to be defined as two 7-element arrays that are indexed from 0. I.e., the type declaration would be:

```
unsigned int CMDL[7];
unsigned int CMDH[7];
```

and software would access ADC0 CMDH1 as `ADC0->CMDH[0]` and ADC0 CMDL7 as `ADC0->CMDL[6]`.

Diagram



Fields

Field	Function
31-27 —	Reserved This read-only field is reserved and always has the value 0.
26-24 NEXT	Next Command Select Selects the next command to be executed after this command completes. Multiple commands can be configured in a scan configuration by linking the next command in a daisy-chain sequence. The command buffer number is not indicative of any particular order and the order of execution is strictly controlled by the NEXT field (for example, a sequence of commands could be CMD2 to CMD1 to CMD3). Unending circular command execution is allowed by setting the NEXT field in the last command in a sequence to the first command in the sequence. It is also allowed for a command to set the next command to itself, resulting in a continuous conversion configuration. Setting the next command to 0x0 causes conversions to terminate at the completion of the command. Lower priority trigger events cannot be serviced until a higher priority triggered command (or sequence of commands) completes. 000b - No next command defined. Terminate conversions at completion of current command. If lower priority trigger pending, begin command associated with lower priority trigger. 001b - Select CMD1 command buffer register as next command. 010b-110b - Select corresponding CMD command buffer register as next command 111b - Select CMD7 command buffer register as next command.
23-20 —	Reserved This read-only field is reserved and always has the value 0.

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
19-16 LOOP	Loop Count Select Selects how many times this command executes (and stores conversion result to RESFIFO) before finish and transition to the next command or Idle state. LWI field controls whether a single channel is converted on each iteration or auto channel increment results in channel scanning functionality. 0000b - Looping not enabled. Command executes 1 time. 0001b - Loop 1 time. Command executes 2 times. 0010b - Loop 2 times. Command executes 3 times. 0011b-1110b - Loop corresponding number of times. Command executes LOOP+1 times. 1111b - Loop 15 times. Command executes 16 times.
15-12 AVGS	Hardware Average Select Selects how many ADC conversions are averaged to create the ADC result (2^{AVGS}). An internal storage buffer is used to capture temporary results while the averaging iterations are executed. Hardware averaging is a nested loop control and does not extend across LOOP boundaries. See Functional description for more detailed description on usage of AVGS, LOOP, and NEXT bitfields in command execution sequencing. 0000b - Single conversion. 0001b - 2 conversions averaged. 0010b - 4 conversions averaged. 0011b - 8 conversions averaged. 0100b - 16 conversions averaged. 0101b - 32 conversions averaged. 0110b - 64 conversions averaged. 0111b - 128 conversions averaged. 1000b - 256 conversions averaged. 1001b - 512 conversions averaged. 1010b - 1024 conversions averaged.
11 —	Reserved This read-only field is reserved and always has the value 0.
10-8 STS	Sample Time Select When programmed to 0b000 the minimum sample time of 3.5 ADCK cycles is selected. When STS is programmed to a non-zero value the sample time is ($3.5 + 2^{STS}$) ADCK cycles. The shortest sample time maximizes conversion speed for lower impedance inputs. Extending sample time allows higher impedance inputs to be accurately sampled. Longer sample times can also be used to lower overall power consumption when command looping and sequencing is configured and high conversion rates are not required.

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	000b - Minimum sample time of 3.5 ADCK cycles. 001b - 3.5 + 2^1 ADCK cycles; 5.5 ADCK cycles total sample time. 010b - 3.5 + 2^2 ADCK cycles; 7.5 ADCK cycles total sample time. 011b - 3.5 + 2^3 ADCK cycles; 11.5 ADCK cycles total sample time. 100b - 3.5 + 2^4 ADCK cycles; 19.5 ADCK cycles total sample time. 101b - 3.5 + 2^5 ADCK cycles; 35.5 ADCK cycles total sample time. 110b - 3.5 + 2^6 ADCK cycles; 67.5 ADCK cycles total sample time. 111b - 3.5 + 2^7 ADCK cycles; 131.5 ADCK cycles total sample time.
7 LWI	Loop with Increment Enables automatic channel incrementing. When 0, the LOOP field selects the number of times the selected channel is converted consecutively. When 1, automatic channel incrementing is enabled and the LOOP field defines how many consecutive channels are converted as part of the command execution. Example 1: LOOP = 8h, LWI = 0b, CMDLn[CTYPE] = 0h, CMDLn[ADCH] = Dh. Convert on channel 13A 9 times. Example 2: LOOP = 8h, LWI = 1b, CMDLn[CTYPE] = 0h, CMDLn[ADCH] = Dh. Run channels 13A-21A each one time. Maximum channel scanning using a single command buffer is defined by the maximum value of the LOOP field (16). 0b - Auto channel increment disabled 1b - Auto channel increment enabled
6-3 —	Reserved This read-only field is reserved and always has the value 0.
2 WAIT_TRIG	Wait for Trigger Assertion before Execution. Controls if commands are automatically executed or if a trigger must be received before execution. If WAIT_TRIG is asserted, wait states will be added before the command until the active trigger is asserted again. When WAIT_TRIG is disabled, each command will be automatically executed when called. 0b - This command will be automatically executed. 1b - The active trigger must be asserted again before executing this command.
1-0 CMPEN	Compare Function Enable After an ADC channel input is sampled and converted and any averaging iterations are performed, the CMDH*a*[CMPEN] field guides operation of the automatic compare function to optionally only store when the compare operation is true. When compare is enabled, the conversion result is compared to the compare value registers (CV*a*[CVH] and CV*a*[CVL]). There are multiple options on command sequencing related to the compare function. See Compare function for a detailed explanation of the compare options.

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	NOTE Not all Command Buffers have implemented the CMPEN field. The CMPEN field is only available in Command Buffers that have a corresponding Compare Value register. 00b - Compare disabled. 01b - Reserved 10b - Compare enabled. Store on true. 11b - Compare enabled. Repeat channel acquisition (sample/convert/compare) until true.

34.6.20 Compare Value Register (CV1 - CV7)

Offset

Register	Offset
CV1	200h
CV2	204h
CV3	208h
CV4	20Ch
CV5	210h
CV6	214h
CV7	218h

Function

The compare value registers (CV*a*) contain values used to compare the conversion result when the compare function is enabled. This register is formatted in the same way as the D field in the RESFIFO registers in different modes of operation for both bit position definition and value format using unsigned or signed 2's complement. There is a direct association of each compare value register to a specific command buffer register (i.e., CV1 is only used during execution of CMD1 command).

NOTE

Not all Command Buffers have an associated Compare Value register. The compare function is only available on Command Buffers that have a corresponding Compare Value register.

When the ADC is actively executing commands, the CV*a* register associated with the active command (CMD*a*) should not be updated. Writes to associated CV*a* register during this time may result in unpredictable behavior.

NOTE

The 7 compare value registers are numbered [CV1] through [CV7]. In NXP-supplied header files, this is likely to be defined as a 7-element array that is indexed from 0. I.e., the type declaration would be:

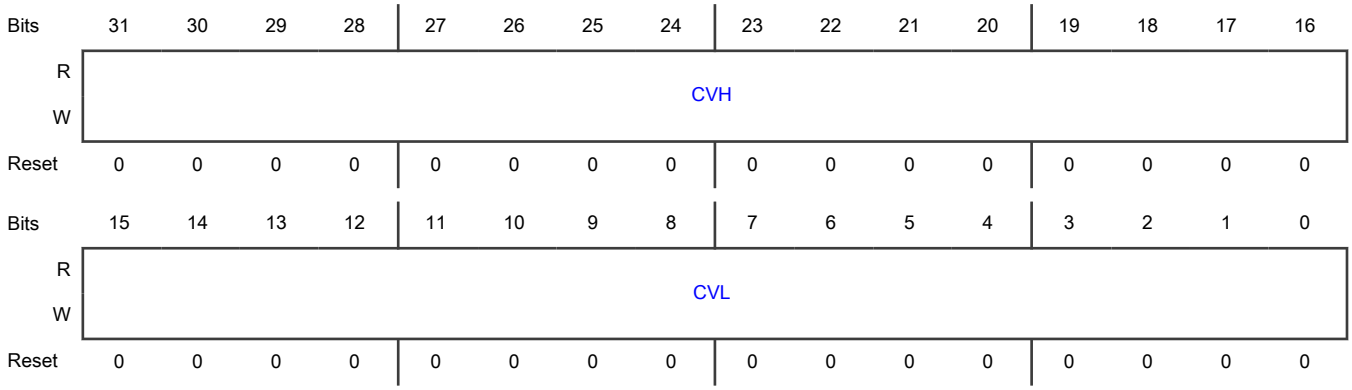
```
unsigned int CV[7];
```

and software would access ADC0 CV1 as `ADC0->CV[0]` and ADC0 CV7 as `ADC0->CV[6]`.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram



Fields

Field	Function
31-16 CVH	Compare Value High The compare function can be configured to check whether the result is less than, greater than, or if the result falls within or outside a range determined by two compare values. After the input is sampled and converted and any averaging iterations are performed, the compare values in CVL and CVH are optionally used in a compare operation on the result. See Compare function for a description on how CVH is used.
15-0 CVL	Compare Value Low The compare function can be configured to check whether the result is less than, greater than, or if the result falls within or outside a range determined by two compare values. After the input is sampled and converted and any averaging iterations are performed, the compare values in CVL and CVH are optionally used in a compare operation on the result. See Compare function for a description on how CVL is used.

34.6.21 Data Result FIFO Register (RESFIFO0)

Offset

Register	Offset
RESFIFO0	300h

Function

The data result FIFO register (RESFIFO) is a 8 entry FIFO that stores the data result of ADC conversions. In addition, several tag fields of source command and trigger information are stored along with the data. FCTRL0[FCOUNT] indicates how many valid datawords are stored in the RESFIFO. Reading RESFIFO provides the oldest unread dataword entry in the FIFO and decrements FCTRL0[FCOUNT]. The FIFO can be emptied by successive reads of RESFIFO. The FIFO is reset by writing 0b1 to the CTRL[RSTFIFOx] bit.

The following table describes the format of data in the result FIFO.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

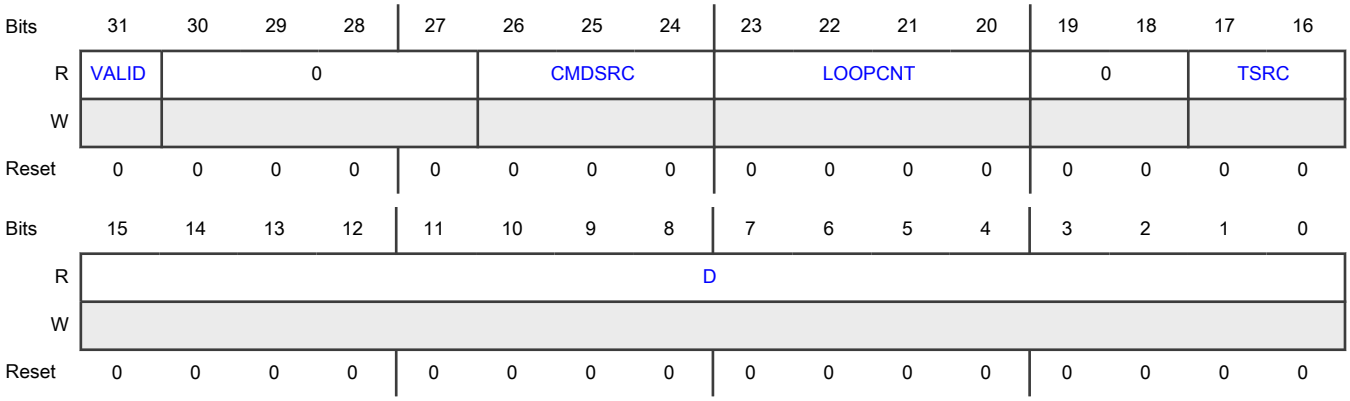
Table 218. Data result register format description

Conversion mode	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	Format
16-bit single-ended	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	Unsigned, 16-bit magnitude, JLEFT=X
12-bit single-ended	0	D	D	D	D	D	D	D	D	D	D	D	D	0	0	0	Unsigned, JLEFT=0, zero in D[15] and D[2:0]
12-bit single-ended	D	D	D	D	D	D	D	D	D	D	D	D	0	0	0	0	Unsigned, JLEFT=1, zero in D[3:0]

NOTE

D: Data

Diagram



Fields

Field	Function
31 VALID	FIFO Entry is Valid Indicate the FIFO entry is valid. When the FIFO is holding data the VALID bit is set. When the FIFO is empty the VALID bit is clear. 0b - FIFO is empty. Discard any read from RESFIFO. 1b - FIFO record read from RESFIFO is valid.
30-27 —	Reserved This read-only field is reserved and always has the value 0.
26-24 CMDSRC	Command Buffer Source Indicate the command buffer being executed that generated this result. 000b - Not a valid value CMDSRC value for a dataword in RESFIFO. 0x0 is only found in initial FIFO state prior to an ADC conversion result dataword being stored to a RESFIFO buffer.

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	001b - CMD1 buffer used as control settings for this conversion. 010b-110b - Corresponding command buffer used as control settings for this conversion. 111b - CMD7 buffer used as control settings for this conversion.
23-20 LOOPCNT	Loop Count Value Indicate the loop count value during command execution that generated this result. When CMDH*a*[LOOP] is non-zero, the command execution stores off a result multiple times during command execution (at the loop boundary). 0000b - Result is from initial conversion in command. 0001b - Result is from second conversion in command. 0010b-1110b - Result is from LOOPCNT+1 conversion in command. 1111b - Result is from 16th conversion in command.
19-18 —	Reserved This read-only field is reserved and always has the value 0.
17-16 TSRC	Trigger Source Indicate the trigger source that initiated a conversion and generated this result. When multiple commands are chained together using the CMDHa[NEXT] field, the TSRC field for all datawords stored to the RESFIFO indicate the trigger source that started the sequence of commands. 00b - Trigger source 0 initiated this conversion. 01b - Trigger source 1 initiated this conversion. 10b - Trigger source 2 initiated this conversion. 11b - Trigger source 3 initiated this conversion.
15-0 D	Data Result The formatting for the data in D is summarized in Table 218.

34.6.22 Calibration General A-Side Registers (CAL_GAR0 - CAL_GAR33)

Offset

For a = 0 to 33:

Register	Offset
CAL_GARa	400h + (a × 4h)

Function

The A-side general calibration value registers (CAL_GAR0 - CAL_GAR33) are used to correct for linearity errors of the A-side converter. All 34 CAL_GAR registers contain calibration information that is automatically updated during the self calibration sequence. See Calibration for more information on completing ADC calibration steps.

All information provided in this document is subject to legal disclaimers.

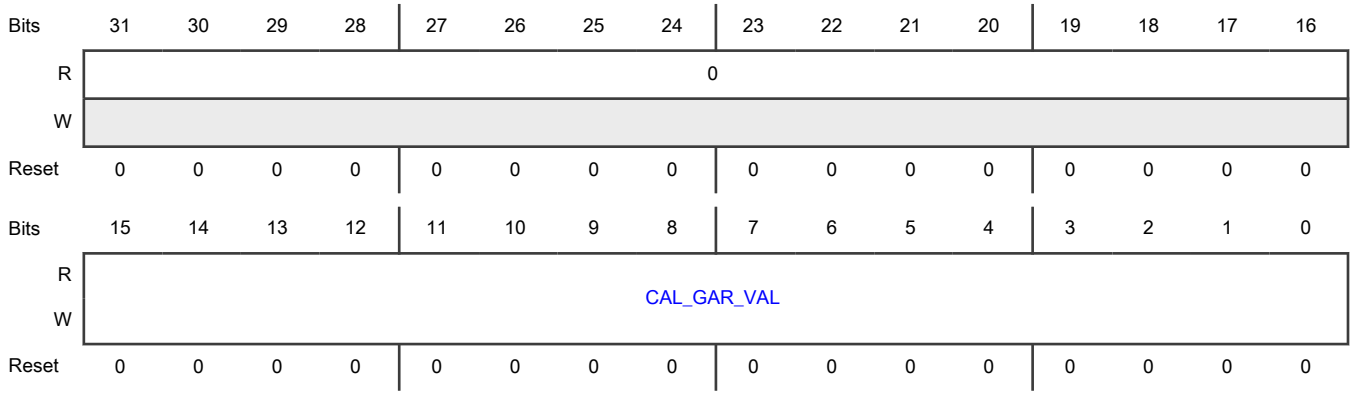
© 2026 NXP B.V. All rights reserved.

The calibration values in the CAL_GAR registers affect the end conversion result by conditionally being subtracted from the conversion before the end result is transferred into the FIFOs. Calibration must be run each time the ADC is powered down or a hard reset is issued. To reduce the latency required to run calibration, the CAL_GAR values can be stored in non-volatile memory after an initial calibration and recovered prior to the first ADC conversion. If these registers are set to values not generated by the calibration function, the linearity error specifications may not be met.

The CAL_GAR registers are only read write accessible when the ADC is disabled with CTRL[ADCEN] = 0b0. Note that access time when writing to these registers is larger than 3 ADC clock cycles. Wait states are inserted on the bus to meet synchronization timing to the associated CAL_GAR register.

Note that the values in each register is a 16-bit signed value but has non-uniform supported range of values. The valid range of each register is summarized in Calibration General A-Side Register valid ranges.

Diagram



Fields

Field	Function
31-16 —	Reserved This read-only field is reserved and always has the value 0.
15-0 CAL_GAR_VAL	Calibration General A Side Register Element The CAL_GAR_VAL is a 16-bit signed value. The valid range for each register in this array is non-uniform and summarized in Calibration General A-Side Register valid ranges.

34.6.23 Configuration 2 Register (CFG2)

Offset

Register	Offset
CFG2	FF8h

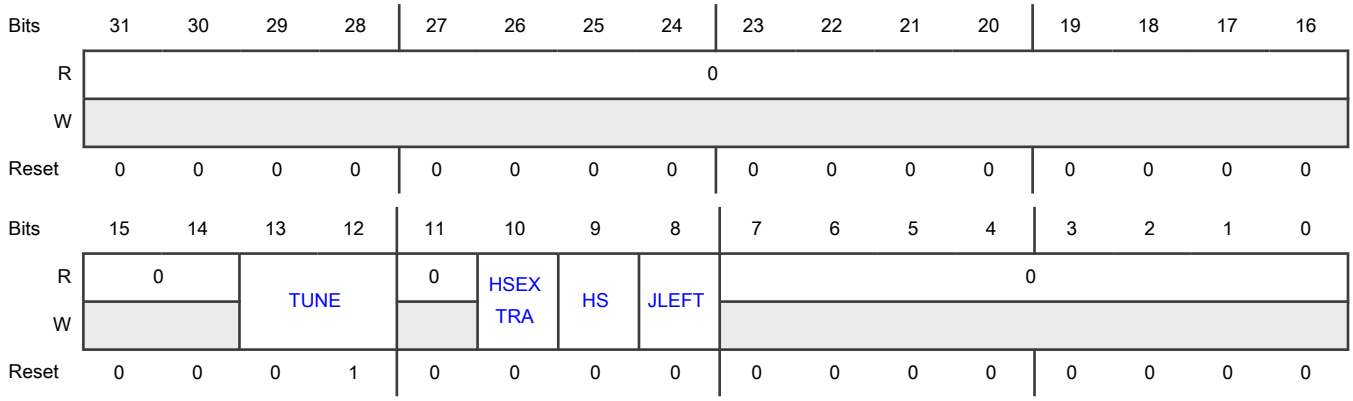
Function

The Configuration 2 Register controls ADC functions that are common to all commands. The CFG2 cannot be changed while the CTRL[ADCEN] bit is set. Writes to CFG2 while CTRL[ADCEN] is set are ignored.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram



Fields

Field	Function
31-14 —	Reserved This read-only field is reserved and always has the value 0.
13-12 TUNE	Tune Mode register TUNE provides some variability in how many cycles are needed to complete a conversion. Conversions can be terminated early at the expense of conversion resolution. See Cycles per conversion for more information on TUNE affect on conversion duration.
11 —	Reserved This read-only field is reserved and always has the value 0.
10 HSEXTRA	High Speed Extra register In high speed conversions (when CFG2[HS] bit set to 1), HSEXTRA provides a secondary control to add an additional ADCK cycle to the conversion. Setting to 1 is needed in a specific corner of operation where VDDA < TBD V and ADCK > TBD Hz. See Cycles per conversion for more information on HSEXTRA affect on conversion duration. 0b - No extra cycle added 1b - Extra cycle added
9 HS	High Speed Enable register Configures ADC for high speed conversions at the expense of higher operating current. When HS is set, conversions complete 2 or 3 ADCK cycles sooner (depending on the CFG2[HSEXTRA] setting) compared to conversion cycle counts when HS is clear. See Cycles per conversion for more information on HS affect on conversion duration. 0b - High speed conversion mode disabled 1b - High speed conversion mode enabled
8 JLEFT	Justified Left Enable register

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	When JLEFT is set and the ADC is converting in 12-bit single-ended mode, conversion result stores to RESFIFOn[D] are in left-justify format. The formatting for the data in RESFIFOn[D] is summarized in Table 218. **NOTE** In JLEFT mode, CVL and CVH must be programmed in left-justify format to generate correct compare results.
7-0 —	Reserved This read-only field is reserved and always has the value 0.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Chapter 35
Operational Amplifier (OPAMP)

35.1 Chip-specific OPAMP information

Table 219. Reference links to related information

Topic	Related module	Reference
Full description	OPAMP	OPAMP
System memory map		Memory map
Clocking		Clock distribution
Power management		Power management
Signal multiplexing	Port control	Signal multiplexing

35.1.1 Module instances

This chip has one instance of the OPAMP module, OPAMP0.

35.1.2 Enable OPAMP

Write 1b to SPC0_SOC_CNTRL[OPAMP0_EN] to enable OPAMP0.

35.1.3 PREF inputs

OPAMP_CTR[PREF] selects positive reference voltage inputs.

Table 220. PREF inputs

OPAMP_CTR[PREF]	Inputs	Pin
00b	OPAMP0_EXT_REF1	P2_3
01b	VDDA/2	—
10b	OPAMP0_EXT_REF0	P2_7
11b	625 mV	—

NOTE

If the OPAMP_CTR[PREF] is configured as 11b, the register SPC.ACTIVE_CFG[BGMODE] must be configured as 10b (Bandgap enabled, buffer enabled).

35.1.4 External signal control

There are analog switches between OPAMP internal signals and package pins of OPAMP0_OUT, OPAMP0_EXT_REF0 and OPAMP0_EXT_REF1 as shown in the following figure. If you want to connect the OPAMP pins with OPAMP internal signals, you must enable the related analog switch.

The analog switch can be enabled by:

- Configure P2_3 to ALT9 OPAMP0_REF_EXT_REF1_EN function
- Configure P2_7 to ALT9 OPAMP0_REF_EXT_REF0_EN function
- Configure P2_16 to ALT9 OPAMP0_OUT_EN function

All information provided in this document is subject to legal disclaimers.
© 2026 NXP B.V. All rights reserved.

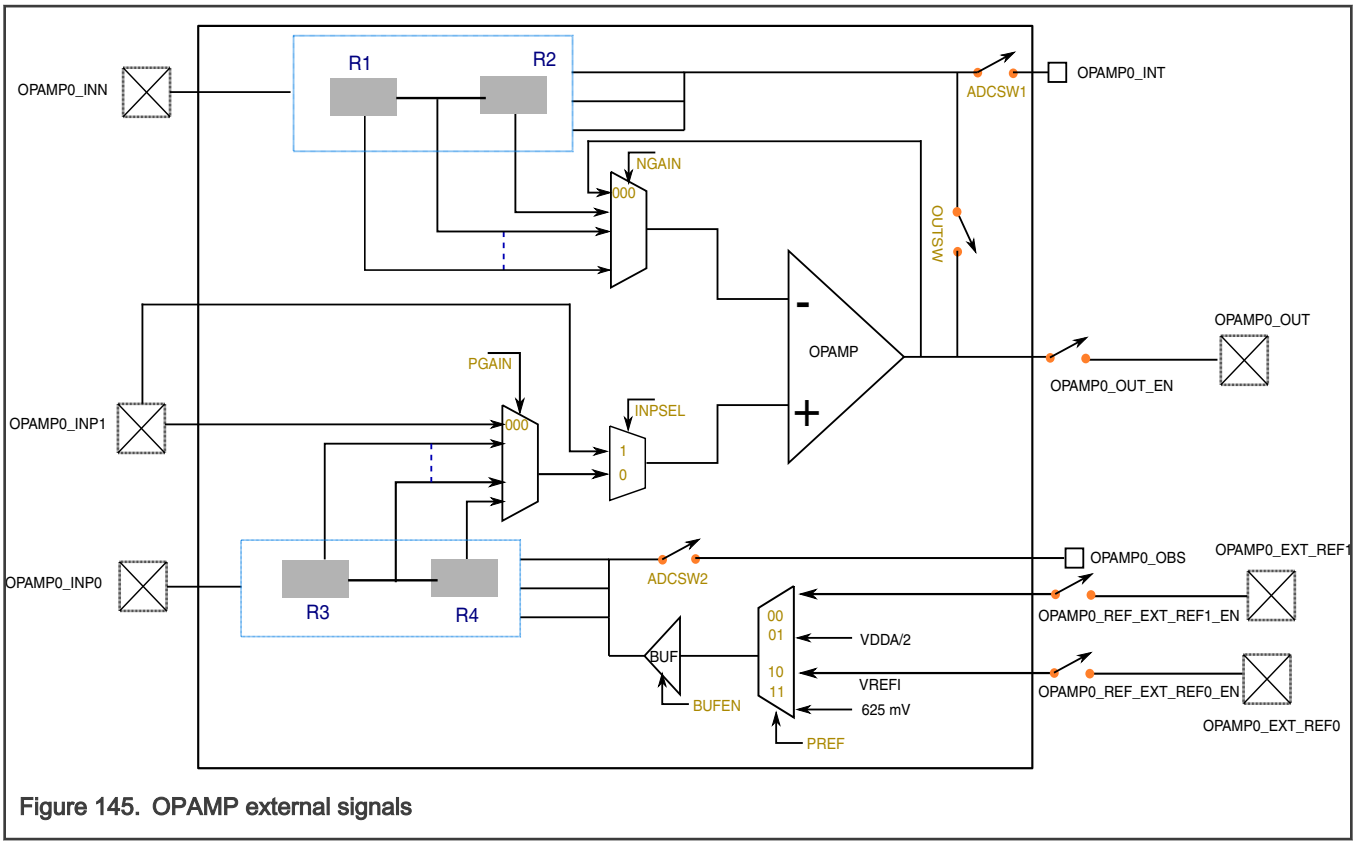


Figure 145. OPAMP external signals

35.2 Overview

The OPAMP is an advanced amplifier that supports programmable gain (PGA), inverting, non-inverting, buffer mode and standalone mode. Positive and negative inputs of an amplifier connect to internal channels. The module is applicable to the signal-processing stage before ADC.

35.2.1 Block diagram

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

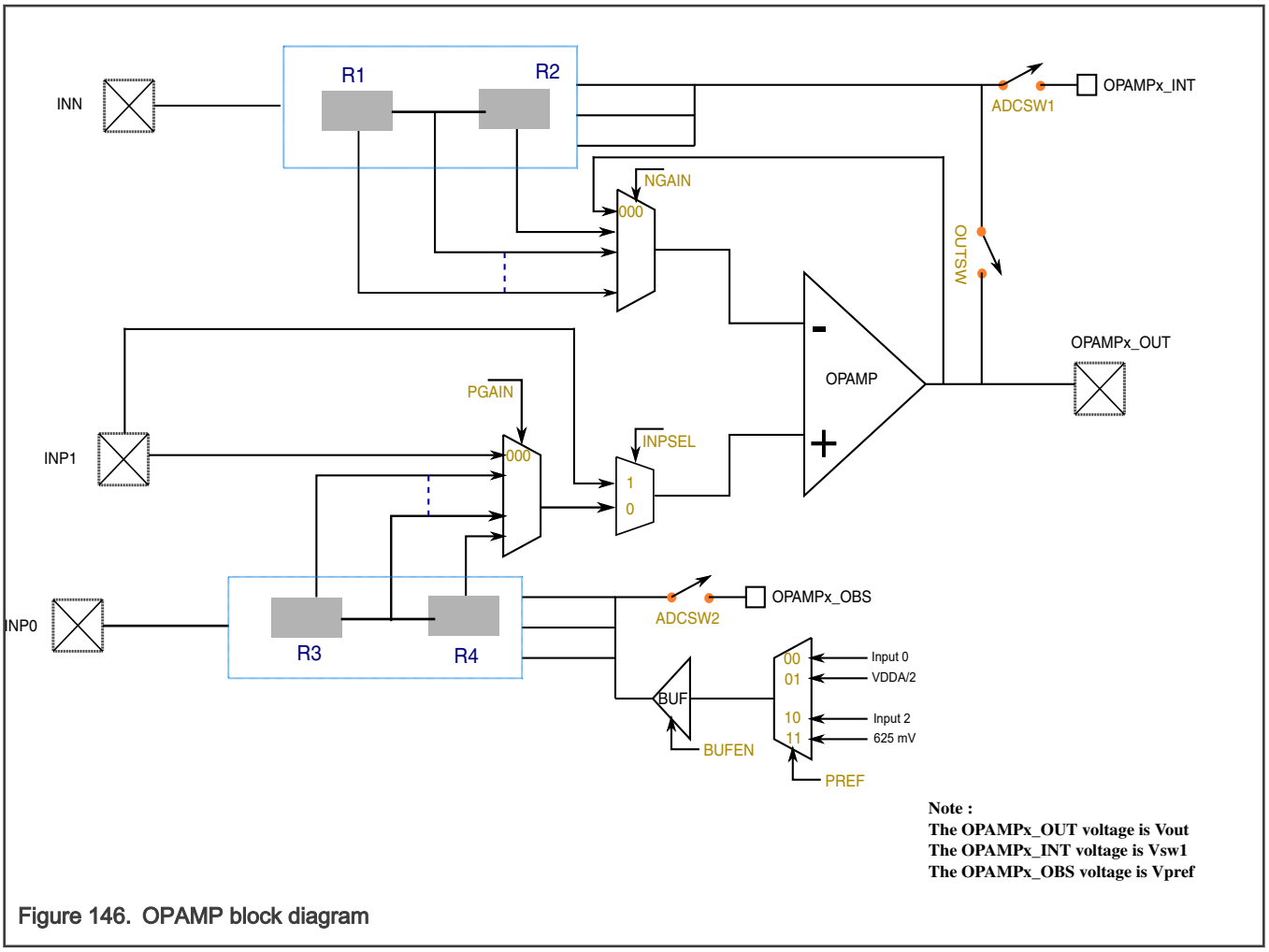


Figure 146. OPAMP block diagram

See chip-specific for supported PREF inputs.

35.2.2 Features

- High performance mode and low power mode
- PGA mode
- Noninverting Amplifier mode
- Inverting Amplifier mode
- Unity gain buffer mode
- Standalone OPAMP mode

35.3 Functional description

This module supports optional gains. See description in OPAMP_CTR[NGAIN] and [PGAIN] for details of programmable gain values where Ngain = R2/R1 and Pgain = R4/R3.

NOTE

The actual gain is determined by different usage methods. For gain calculation examples, see Application Information.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

35.3.1 Modes

This module supports high performance mode and low power mode selected by configuring OPAMP_CTR[MODE].

The table below shows all supported modes and the related configurations:

Table 221. OPAMP modes configuration

Mode	ADCSW1	OUTSW	ADCSW2	NGAIN	PGAIN	INPSEL	PREF	BUFEN
PGA	ON	ON	ON	x	x	INP0	00b/01b/10b/11b	ON
Noninverting	ON	ON	OFF	x	x	INP0, INP1	-	OFF
Inverting[1]	ON	ON	OFF	x	0	INP1	-	OFF
Standalone	OFF	OFF	OFF	011b	011b	INP0, INP1	-	OFF
Buffer	OFF	OFF	OFF	0	0	INP1	-	OFF

1. Vinp0 must be floating

35.3.1.1 PGA Mode

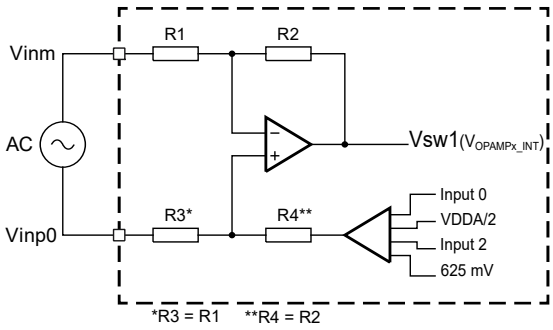


Figure 147. PGA Mode

See chip-specific for supported PREF inputs.

In this mode, configure the registers OPAMP_CTR[NGAIN] and OPAMP_CTR[PGAIN] to select different gains. Enable OPAMP_CTR[BUFEN] and configure the registers OPAMP_CTR[PREF] to select positive reference voltage. Configure OPAMP_CTR[OUTSW] to connect R2 with OPAMP output. Enable OPAMP_CTR[ADCSW1] and OPAMP_CTR[ADCSW2], measure the results from OPAMPx_INT and OPAMPx_OBS by ADC.

35.3.1.2 Noninverting Amplifier mode

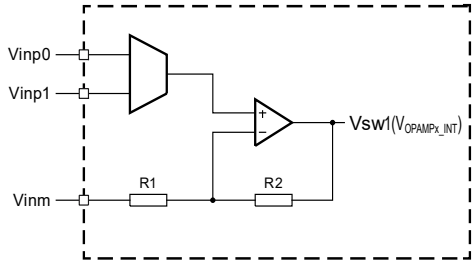


Figure 148. Noninverting Amplifier mode

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

This mode output a forward gain voltage and the output voltage is connected to the ADC. Configure the registers **OPAMP_CTR[NGAIN]** to select different gains. The positive input can be from Vinp0 or Vinp1. If Vinp1 is used, configure **OPAMP_CTR[PGAIN]** to 0. Configure OPAMP_CTR[BUFEN] to 0 to disable reference buffer. Enable OPAMP_CTR[OUTSW] to connect R2 with OPAMP output (OPAMPx_OUT). Enable OPAMP_CTR[ADCSW1], and measure results from OPAMPx_INT.

35.3.1.3 Inverting Amplifier Mode

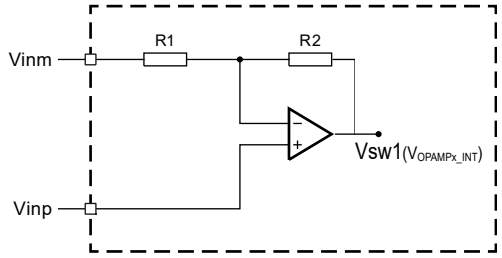


Figure 149. Inverting Amplifier Mode

This mode output a reverse gain voltage and the output voltage is connected to the ADC. Configure the registers **OPAMP_CTR[NGAIN]** to select different gains. The positive input can be either from Vinp1 or reference buffer. Configure **OPAMP_CTR[PGAIN]** to 0 to select Vinp1. Enable reference buffer by **OPAMP_CTR[BUFEN]** to select reference buffer as the source of positive input. Vinp0 must be floating. Enable **OPAMP_CTR[OUTSW]** to connect R2 with OPAMP output (OPAMPx_OUT). Enable **OPAMP_CTR[ADCSW1]**, and measure results from OPAMPx_INT.

35.3.1.4 Standalone OPAMP Mode

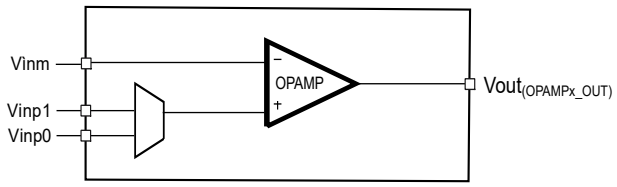


Figure 150. Standalone OPAMP Mode

To use standalone OPAMP mode, disconnect R2 with OPAMP output by disabling **OPAMP_CTR[OUTSW]**. Configure **OPAMP_CTR[NGAIN]=011b** and **OPAMP_CTR[PGAIN]=011b**. Disable reference buffer by **OPAMP_CTR[BUFEN]**. Disconnect ADCSW1 and ADCSW2.

35.3.1.5 Buffer Mode

To select buffer mode, configure the value of registers **OPAMP_CTR[NGAIN]** and **OPAMP_CTR[PGAIN]** to 000b. The output voltage is connected to the **OPAMPx_OUT**. Configure **OPAMP_CTR[BUFEN]** to 0 to disable reference buffer. Enable **OPAMP_CTR[OUTSW]** to connect R2 with OPAMP output. Enable **OPAMP_CTR[ADCSW1]**, and measure results from **OPAMPx_INT**.

35.3.2 Clock

The global clock is used for OPAMP.

35.3.3 Reset

OPAMP is reset by the global chip reset signal.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

35.3.4 Interrupts

This module has no interrupts.

35.4 External Signals

Table 222. External Signal Description

Signal	Description	I/O
INN	Analog input channels(Core amplifier negative input channel)	I
INP0	Analog input channels(Core amplifier positive input channel0)	I
INP1	Analog input channels(Core amplifier positive input channel1)	I
V_{OPAMPx_OUT}	Output of the core amplifier	O

35.5 Initialization

This module does not require initialization.

35.6 Application Information

Below is the OPAMP output equation :

$$V_{sw1} = \frac{Ngain+1}{1+\frac{1}{Pgain}} V_{INP0} - Ngain * V_{INN} + \frac{1+Ngain}{1+Pgain} V_{pref}$$

35.6.1 PGA mode Application Information

- Both OUTSW and ADCSW1/2 switches should be on.
- OPAMP_CTR[NGAIN] and OPAMP_CTR[PGAIN] should be set to same value.
- Measurement method:
  - Two steps method
    1. Measure OPAMPx_OBS voltage, defined as Vpref (voltage input selected by PREF bit) first with ADC single end mode, and the measured Vpref will be used in following calculation.
    2. Measure OPAMPx_INT voltage, defined as Vsw1 with ADC single end mode real time.
  - Differential method
    1. Use ADC differential channel measure Vsw1 and Vpref at same time.
- ADCSW1 could be on to fine-tune the results
- Result

  Recommend Ngain=Pgain,
  - Vsw1 –Vpref = (Vinp0-Vinm) * Ngain

35.6.2 Noninverting mode Application Information

- ADCSW1 should be closed
- Disable the OPAMP_CTR[BUFEN]

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

- Result
  - Vsw1 + Ngain * Vinm= (1+Ngain) * Vinp

  NXP recommends you to connect Vinm to GND, if you do not need to provide a DC offset, and Vinp is recommended to be connected to Vinp1.

35.6.3 Inverting mode Application Information

- ADCSW1 should be closed
- Disable the OPAMP_CTR[BUFEN]
- Result
  - Vsw1 = (Ngain+1) * Vinp-Ngain * Vinm
- OPAMP_CTR[PGAIN] is recommended to be set to 000b.

  NXP recommends you to connect Vinp to Vinp1. The recommended voltage of Vinp0/1 is half of VDD_ANA. Vinp cannot be 0.

35.6.4 Buffer mode Application Information

- The unity gain buffer mode supports OPAMPx_OUT or OPAMPx_INT as the output, with Vout defined as the voltage.
- Disable the OPAMP_CTR[BUFEN]
- Result
  - Vout=Vinp1

  Vinp is recommended to be connected to Vinp1.

35.7 Memory map and register definition

This section includes the OPAMP module memory map and detailed descriptions of all registers. You can select different gains by configuring registers. Registers have options for selecting noninverting and inverting gain application.

35.7.1 OPAMP register descriptions

35.7.1.1 OPAMP memory map

OPAMP0 base address: 400B_7000h

Offset	Register	Width (In bits)	Access	Reset value
0h	Version ID (VERID)	32	R	0000_0000h
4h	Parameter (PARAM)	32	R	0000_0001h
8h	OPAMP Control (OPAMP_CTR)	32	RW	0100_0000h

35.7.1.2 Version ID (VERID)

Offset

Register	Offset
VERID	0h

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Function

Version ID indicates the version integrated for this instance on the chip. It also indicates inclusion/exclusion of several optional features.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	MAJOR								MINOR							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	FEATURE															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-24 MAJOR	Major Version Number Returns the major version number for the specification.
23-16 MINOR	Minor Version Number Returns the minor version number for the specification.
15-0 FEATURE	Feature Specification Number Returns the feature set number.

35.7.1.3 Parameter (PARAM)

Offset

Register	Offset
PARAM	4h

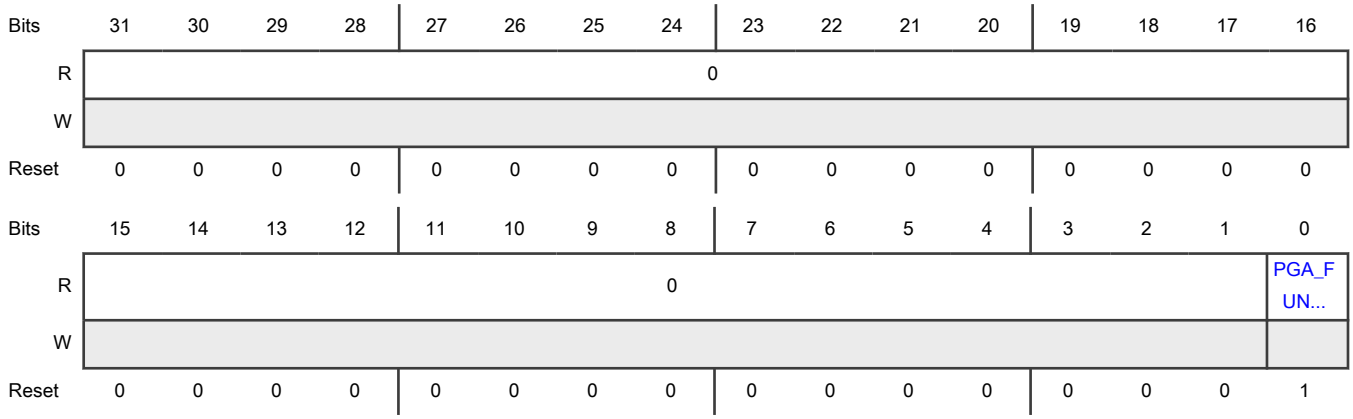
Function

PARAM indicates optional PGA functions or core amplifier for this instance on the chip.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram



Fields

Field	Function
31-1 —	Reserved
0 PGA_FUNCTION	PGA Function Option Selects different PGA functions. 0b - Core amplifier enabled 1b - PGA function enabled

35.7.1.4 OPAMP Control (OPAMP_CTR)

Offset

Register	Offset
OPAMP_CTR	8h

Function

OPAMP_CTR is used to configure the OPAMP module.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	NGAIN			0	PGAIN			0	OUTSW	ADCSW2	ADCSW1	0	PREF		BUFEN
W																
Reset	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0			INPF	0		INPSEL	Reserved	0		INTREF		BIASC		MODE	EN
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31 —	Reserved
30-28 NGAIN	Negative PGA Selection Selects negative programmable gains. 000b - Buffer 001b - 0.95 010b - 1.85 011b - 3.55 100b - 6.55 101b - 11.3 110b - 17.9 111b - 24.6
27 —	Reserved
26-24 PGAIN	Positive PGA Selection Selects positive programmable gains. 000b - Positive input 1 (INP1) 001b - 0.95 010b - 1.85 011b - 3.55 100b - 6.55 101b - 11.3

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	110b - 17.9 111b - 24.6
23 —	Reserved
22 OUTSW	Output Switch Provides output of OPAMP to the negative gain resistor ladder switch. 0b - OPAMP out to negative gain resistor ladder switch off 1b - OPAMP out to negative gain resistor ladder switch on
21 ADCSW2	Measure Switch 2 Specifies whether the measure positive gain resistor ladder reference voltage switch is on or off for ADC channel. 0b - Measure positive gain resistor ladder reference voltage switch off 1b - Measure positive gain resistor ladder reference voltage switch on
20 ADCSW1	Measure Switch 1 Specifies whether the measure negative gain resistor ladder voltage switch is on or off for ADC channel. 0b - Measure negative gain resistor ladder voltage switch off 1b - Measure negative gain resistor ladder voltage switch on
19 —	Reserved
18-17 PREF	Positive Reference Voltage Selection Selects positive reference voltage. For inputs 0 and 2, see chip-specific information for supported PREF inputs. 00b - Input 0 01b - VDDA/2 10b - Input 2 11b - 625 mV
16 BUFEN	Reference Buffer Enables reference buffer. 0b - Disables 1b - Enables
15-13	Reserved

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
—	
12 INPF	Positive Input Connection Status The channel which is connected to OPAMP positive input. 0b - Positive input 0 (INP0) 1b - Positive input 1 (INP1)
11-10 —	Reserved
9 INPSEL	Positive Input Channel Selection Selects the positive input channel. 0b - Selects positive input 0 (INP0) 1b - Selects positive input 1 (INP1)
8 —	**NOTE** This reserved bit must be kept as 0. Writing to it can cause unexpected behavior.
7-6 —	Reserved
5-4 INTREF	Provide OPAMP rail to rail voltage selection 00b - Select OPAMP input rail to rail voltage from 0 to VDD_ANA 01b - Select OPAMP input rail to rail voltage from 0 to VDD_ANA-0.8V 10b - Select OPAMP input rail to rail voltage from 0.8V to VDD_ANA 11b - Not allowed
3-2 BIASC	Bias Current Trim Selection 00b - Default 01b - Increase the current by 10%, total 110% of default value 10b - Decrease the current by 10%, total 90% of default value 11b - Decrease the current by 20%, total 80% of default value
1 MODE	Mode Selection Selects one mode between High performance mode and Low power mode. 0b - High performance mode 1b - Low power mode
0 EN	OPAMP Enable Enables OPAMP.

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	0b - Disable 1b - Enable

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Chapter 36
Low Power Comparator (LPCMP)

36.1 Chip-specific LPCMP information

Table 223. Reference links to related information

Topic	Related module	Reference
Full description	LPCMP	LPCMP
System memory map		Memory map
Clocking		Clock distribution
Power management		Power management
Signal multiplexing	Port control	Signal multiplexing

36.1.1 Module instances

This chip has one instance of the LPCMP module, LPCMP0.

36.1.2 Input connections

Table 224. Input connections

Input channel	CCR2_PSEL[2:0]	INP	CCR2_MSEL[2:0}	INM
0	000b	CMP0_IN0	000b	CMP0_IN0
1	001b	CMP0_IN1	001b	CMP0_IN1
2	010b	Reserved	010b	Reserved
3	011b	CMP0_IN3	011b	CMP0_IN3
4	100b	OPAMP0_OUT/ CMP0_IN4P	100b	Reserved
5	101b	OPAMP0_INT	101b	Reserved
6	110b	Reserved	110b	Reserved
7	111b	LPCMP0 DAC	111b	LPCMP0 DAC

36.1.3 Input references

In the DAC Control register, when:

- DCR[VRSEL]=0, VREFH0 is selected, which is VDD
- DCR[VRSEL]=1, VREFH1 is selected, which is VREFI

36.1.4 Trigger sources

RRCR0[RR_TRG_SEL] controls the trigger source for the LPCMP. When this field is configured as:

- 0 - External trigger of LPTMR0 is selected as the trigger source.
- 1 - Internal trigger

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

36.2 Overview

The LPCMP module provides a circuit to compare two analog input voltages. It includes the following:

- A low power comparator (CMP)
- A DAC
- An analog mux (ANMUX)

See Block diagram for more information.

LPCMP can operate across the full range of the supply voltage, known as rail-to-rail operation.

DAC is a 256-tap resistor ladder network that provides a selectable voltage reference for applications requiring a voltage reference. DAC divides the supply reference V_{in} into 256 voltage levels. An 8-bit digital signal input selects the output voltage level, which varies from V_{in} to $V_{in}/256$.

You can select V_{in} from the following voltage sources:

- VREFH0
- VREFH1

See the Chip-specific LPCMP information for more information on source of VREFH0 and VREFH1.

NOTE

The LPCMP's internal DAC output is available as an on-chip internal signal only and is not available for an external chip pin.

ANMUX allows you to select an analog input signal from among eight channel options. One channel option is the DAC output. Other chip resources are connected to the other channels. See the Chip-specific LPCMP information section for more information. ANMUX can operate across the full range of the supply voltage.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

36.2.1 Block diagram

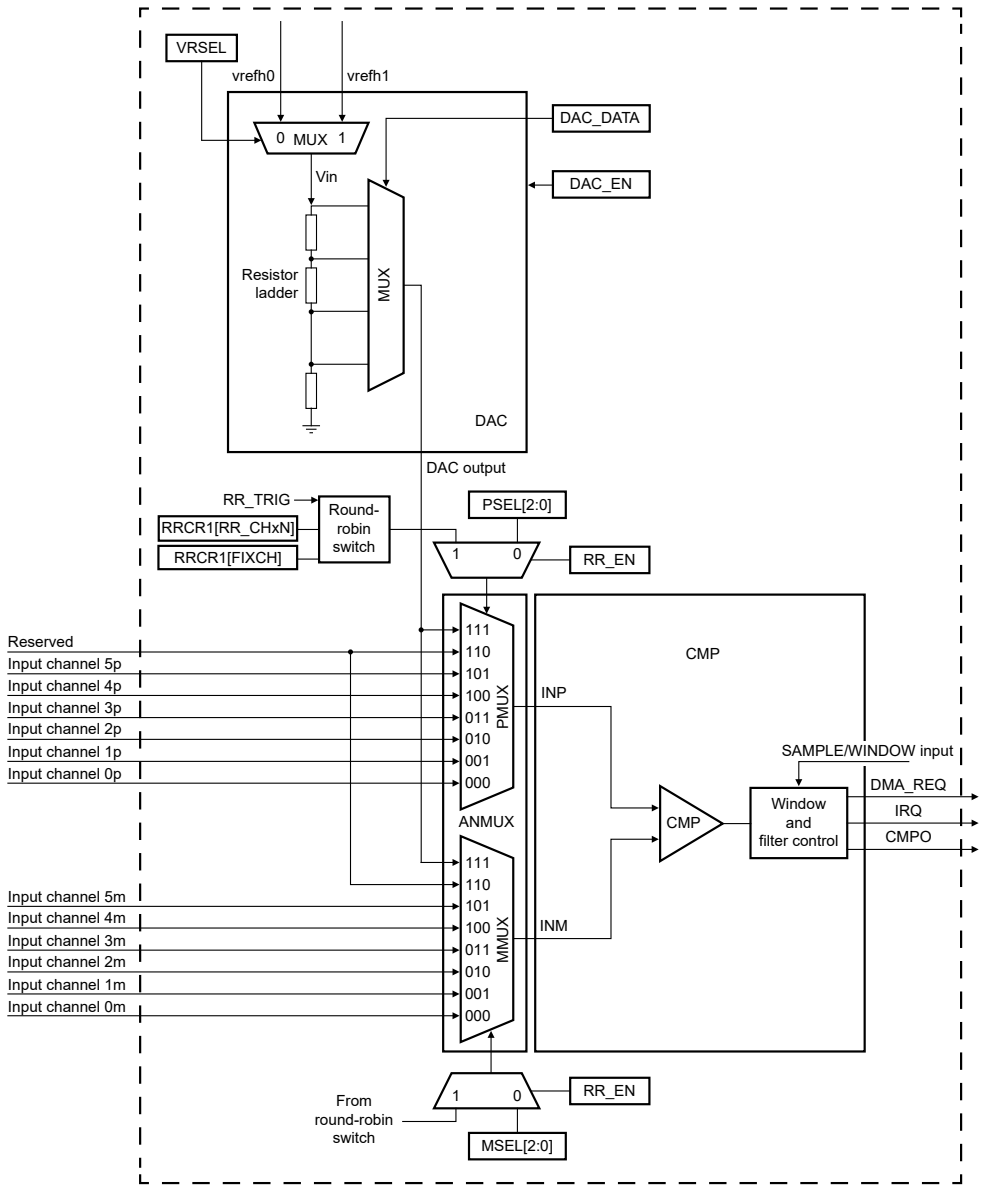


Figure 151. Block diagram

36.2.2 Features

The features of the LPCMP module include:

- Includes two 8-to-1 channel MUXes to select input signal from eight channels
- Supports multiple operation modes to produce a wide range of outputs such as:
  - Sampled
  - Windowed, which is ideal for certain PWM zero-crossing-detection applications
  - Digitally filtered
- Provides the following advance features for window and sample:

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

- Window and sample signals can be inverted.
- COUT (comparator output) rising, falling or both edges closes the window.
- COUT level can be defined when window is closed.

- Provides selectable performance levels:
  - Nano-Power mode
  - Low-Power (speed) mode
  - High-Power (speed) mode
- Supports programmable hysteresis control
- Provides a selectable inversion on comparator output
- Uses an external hysteresis at the same time the output filter is used for internal functions
- Provides interrupt and DMA support
- Supports Round Robin Trigger mode
- Includes an 8-bit resolution DAC
- Provides a selectable supply reference source for DAC
- Provides a configurable Low-Power (speed) mode or High-Power (speed) mode for DAC

36.3 Functional description

36.3.1 Functional block diagram

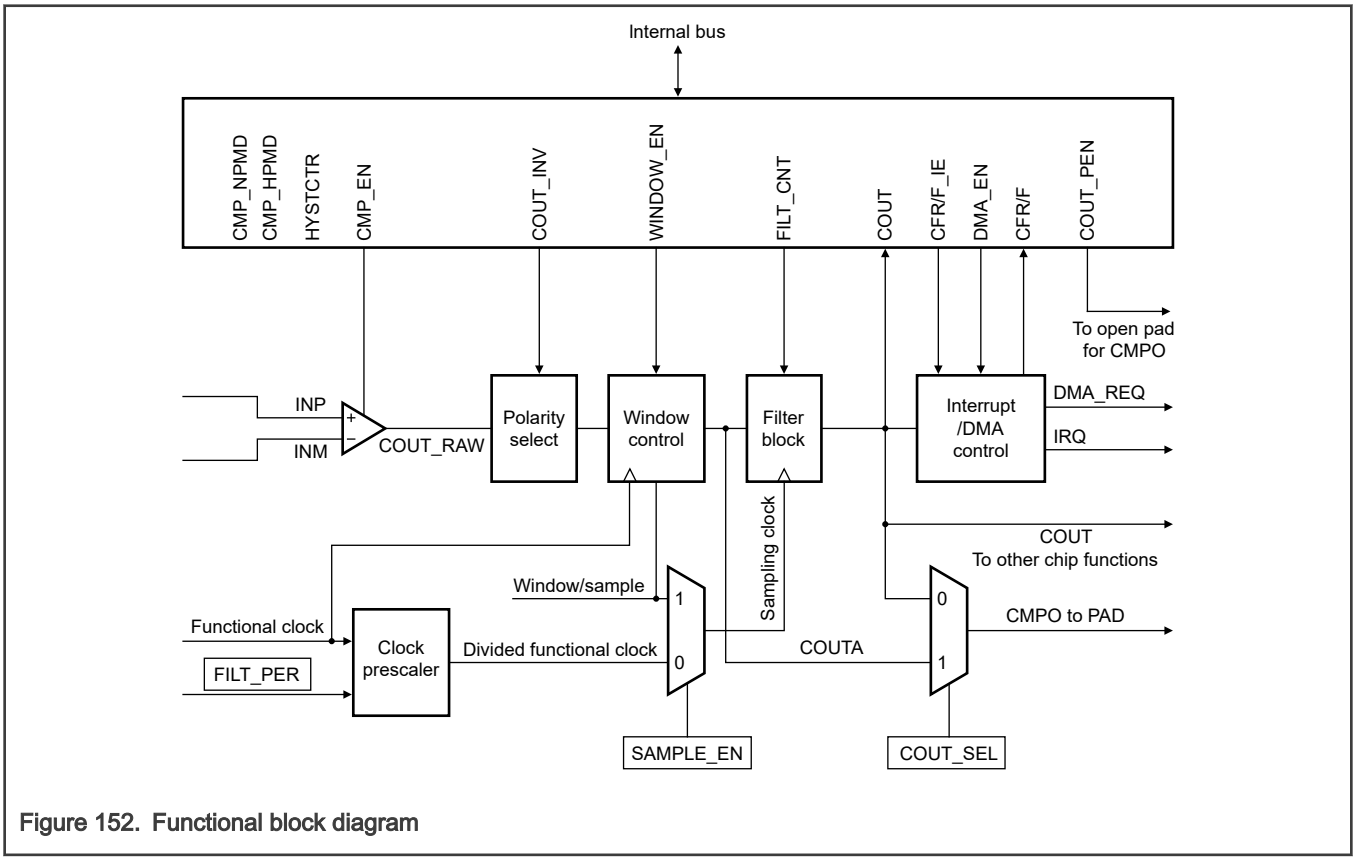


Figure 152. Functional block diagram

As shown in the block diagram, the functions are:

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

- Compared two analog input voltages applied to INP and INM, COUT_RAW is high when the INP input voltage is greater than the INM input voltage, and COUT_RAW is low when the INP input voltage is less than the INM input voltage.
- The COUT_RAW signal can be inverted by enabling CCR1[COUT_INV].
- The optionally inverted comparator output COUT_RAW is sampled on every functional clock when you enable the CCR1[WINDOW_EN] to generate COUTA. In this case, the comparator output is ignored during time periods when the input voltages are not valid. This is useful when you implement zero-crossing-detection for certain PWM applications.
- The window control block is bypassed when CCR1[WINDOW_EN] is disabled.
- The filter block acts as a simple sampler when CCR1[FILT_CNT] is set to 01h.
- The filter block acts as a filter based on multiple samples when CCR1[FILT_CNT] is set to be greater than 01h.
  - If CCR1[SAMPLE_EN] is set to 1, use the external SAMPLE input as the sampling clock.
  - If CCR1[SAMPLE_EN] is set to 0, use the divided functional clock as the sampling clock.
- Bypasses the filter block when it is not in use.

```
Bypass_Filter_Block = (FILT_CNT == 0x00) |  (~SAMPLE_EN & (FILT_PER == 0x00))
```

- Both COUTA and COUT can be configured as module output CMPO by configuring CCR1[COUT_SEL], and are used for different purposes within the system.
- The optionally filtered COUT can be read directly in CSR[COUT].
- The SAMPLE/WINDOW signal can be inverted by setting CCR1[WINDOW_INV].
- The SAMPLE/WINDOW signal can be closed by COUT's falling edge and/or rising edge by setting CCR1[WINDOW_CLS] in Window mode.
- In Window mode, when window is closed, define the COUTA value as CCR1[COUTA_OW] by setting CCR1[COUTA_OWEN]. If CCR1[COUTA_OWEN] is not set, COUTA holds the last sampled value.

NOTE

See the chip configuration section for the source of SAMPLE/WINDOW input.

36.3.2 Round-robin trigger mode

You can enable Round-Robin Trigger mode by setting RRCR0[RR_EN] to 1. A trigger event initiates a comparison sequence. The next trigger event should not occur before the current sequence completes.

RRCR1[FIXP] and RRCR1[FIXCH] select the reference channel for the plus side mux or the minus side mux. RRCR1[RR_CHnEN] selects active channels.

When a trigger comes, the analog comparator enables. After the comparison sequence completes, the analog comparator disables again. RRCR0[RR_INITMOD] controls the analog stabilization time.

NOTE

RR_INITMOD*round robin clock period must be longer than the initialization delay specified in the Comparator and 8-bit DAC electrical specifications section of LPCMP datasheet.

After the stabilization process completes, the round robin manner comparison sequence begins. Sample the comparison result for the selected active channel after RRCR0[RR_NSAM] defines the configurable number of operation clocks.

There are (RR_SAMPLE_CNT + 1) samples for each channel. With these samples, at least (RR_SAMPLE_THRESHOLD + 1) sampled "1" causes the channel's final sample result to be "1"; otherwise, it is "0". Compare the final result with the pre-programmed value.

After all the active channels are sampled/compared, if the comparison result changes from its pre-programmed state, the corresponding flag in RRSR[RR_CHnF] is set. Write to RRCSR[RR_CHnOUT] to configure the pre-programmed state for each channel. Update RRCSR[RR_CHnOUT] to store the last comparison result for each channel. If any flag in RRSR[RR_CHnF] sets,

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

CSR[RRF] also sets. If IER[RRF_IE] sets, an asynchronous interrupt asserts. Note that these flags do not support generating a DMA transfer event.

The following diagram shows the basic flow of this mode. In the diagram, RRCR1[RR_CH1EN], RRCR1[RR_CH3EN], and RRCR1[RR_CH4EN] are 1, so channels #1, #3, and #4 are selected for round-robin depending on their priority setting. RRCR0[RR_NSAM] sets to 2'b01, so you can sample one clock later the comparison result of the selected channel. After you compare the channel #4, the result is sampled, and round-robin ends. If any of the comparison results from channel #1, #3, or #4 changed from their programmed value (written to RRCSR[RR_CH1OUT], RRCSR[RR_CH3OUT], and RRCSR[RR_CH4OUT]), generates an interrupt. Software can then poll RRSR[RR_CHnF] to see which channel input(s) changed value.

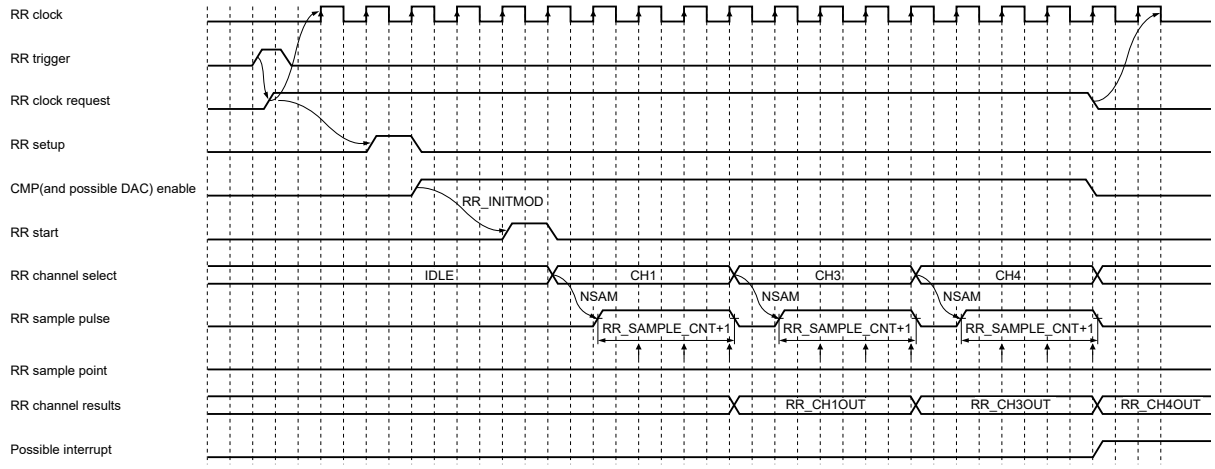


Figure 153. Trigger mode

The table below shows the channel decode in both Functional mode and Trigger mode. Other cases not in the table are illegal.

Table 225. CMP channel decode in functional mode and round-robin trigger mode

Mode	RR_EN	PSEL[2:0]	MSEL[2:0]	FIXP	FIXCH[2:0]	RR_CHxN	INP	INM	CMP Behavior
Functional mode	0	0 to 7	0 to 7	x[1]	x	x	Channel decoded from PSEL[2:0]	Channel decoded from MSEL[2:0]	Compare the channel 0 to 7 with channel 0 to 7[2]
Trigger mode	1	x	x	0	0 to 7	0 to 7	Channel fixed by FIXCH[2:0]	Channel sweep (RR_CHxN)	Sweep the channel 0 to 7 with a fixed channel (0 to 7)
		x	x	1	0 to 7	0 to 7	Channel sweep (RR_CHxN)	Channel fixed by FIXCH[2:0]	Sweep the channel 0 to 7 with a fixed channel (0 to 7)

1. "x" means "don't care"
2. PSEL should not be same as MSEL.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

36.3.3 Low-pass filter mode

The low-pass filter mode operates on an unfiltered, optionally inverted comparator output COUTA, and generates the filtered and synchronized output COUT. You can configure both COUTA and COUT as module outputs and use for different purposes within the system.

Synchronization and edge detection determine the bit values of status register. They also apply to COUT for all sampling and windowed modes. You can perform filtering using an internal timebase defined by CCR1[FILT_PER], or use an external sample input to determine sample time.

The need for digital filtering and the amount of filtering depends on your requirements. Filtering can become more useful in the absence of an external hysteresis circuit. Without external hysteresis, generate a high-frequency oscillations at COUTA when the selected INM and INP input voltages differ by less than the offset voltage of the differential comparator.

36.3.3.1 Enabling low-pass filter mode

You can enable low-pass filter mode by setting the following:

- CCR1[FILT_CNT] > 01h
- CCR1[FILT_PER] to a nonzero value or writing 1 to CCR1[SAMPLE_EN].

If you use the divided functional clock to drive the low-pass filter, it samples COUTA every CCR1[FILT_PER] functional clock cycle.

If CCR1[SAMPLE_EN] is set to 1, the low-pass filter samples COUTA on each positive transition of the sample input. The output state of the filter changes when all the consecutive CCR1[FILT_CNT] samples agree that the output value has changed.

36.3.3.2 Latency issues

Program the value of CCR1[FILT_PER] or sample period such that the sampling period is longer than the period of the expected noise, ensuring that a given noise spike corrupts only one sample. You must choose the value of CCR1[FILT_CNT] to reduce the probability of noisy samples causing an incorrect transition to recognize. The probability of an incorrect transition is defined as the probability of an incorrect sample raised to the power of CCR1[FILT_CNT].

You must trade off the values of CCR1[FILT_PER] or sample period and CCR1[FILT_CNT] against the need for minimal latency in recognizing actual comparator output transitions. The probability of detecting an actual output change within the nominal latency is the probability of a correct sample raised to the power of CCR1[FILT_CNT].

Table 227 summarizes maximum latency values for the various modes of operation in the absence of noise. Filtering latency restarts each time the noise masks an actual output transition.

36.3.4 Low power mode operation

Below table introduces the mode of operation of lower power.

Table 226. Low power mode operation

Mode of operation	Description
Wait	LPCMP can operate normally with window and filter function.
Deep Sleep	LPCMP can operate only in Continuous mode or Round-robin trigger mode.

36.3.5 Functional modes

You can combine the comparator window and filter features as shown in the following table.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table 227. Functional modes

Mode #	CMP_EN	WINDOW_EN	SAMPLE_EN	FILT_CNT	FILT_PER	Operation	Maximum latency[1]
1	0	X	X	X	X	See the Disabled mode (#1).	N/A
2A	1	0	X	0x00	X	See the Continuous mode (#2A and #2B).	T_{PD}
2B	1	0	0	X	0x00		
3A	1	0	1	0x01	X	See the Sampled, non-filtered mode (#3A and #3B).	$T_{PD} + T_{SAMPLE} + T_{per}$
3B	1	0	0	0x01	> 0x00		$T_{PD} + (FILT\_PER * T_{per}) + T_{per}$
4A	1	0	1	> 0x01	X	See the Sampled, filtered mode (#4A and #4B).	$T_{PD} + (FILT\_CNT * T_{SAMPLE}) + T_{SAMPLE}$
4B	1	0	0	> 0x01	> 0x00		$T_{PD} + (FILT\_CNT * FILT\_PER \times T_{per}) + T_{per}$
5A	1	1	0	0x00	X	See the Windowed mode (#5A and #5B).	$T_{PD} + 2T_{per}$
5B	1	1	0	X	0x00		
6	1	1	0	0x01	> 0x00	See the Windowed/ Resampled mode (#6).	$T_{PD} + (FILT\_PER * T_{per}) + 3T_{per}$
7	1	1	0	> 0x01	> 0x00	See the Windowed/ Filtered mode (#7).	$T_{PD} + (FILT\_CNT * FILT\_PER \times T_{per}) + 3T_{per}$
All other combinations of CMP_EN, WINDOW_EN, SAMPLE_EN, FILT_CNT, and FILT_PER are illegal.							

1. T_{PD} represents the intrinsic delay of the analog component plus the polarity select logic. T_{SAMPLE} is the clock period of the external sample clock. T_{per} is the period of the functional clock.

36.3.5.1 Disabled mode (#1)

In this mode:

- The analog comparator is non-functional and consumes no power.
- CSR[COUT] and CMPO are the same as CCR1[COUT_INV].

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

36.3.5.2 Continuous mode (#2A and #2B)

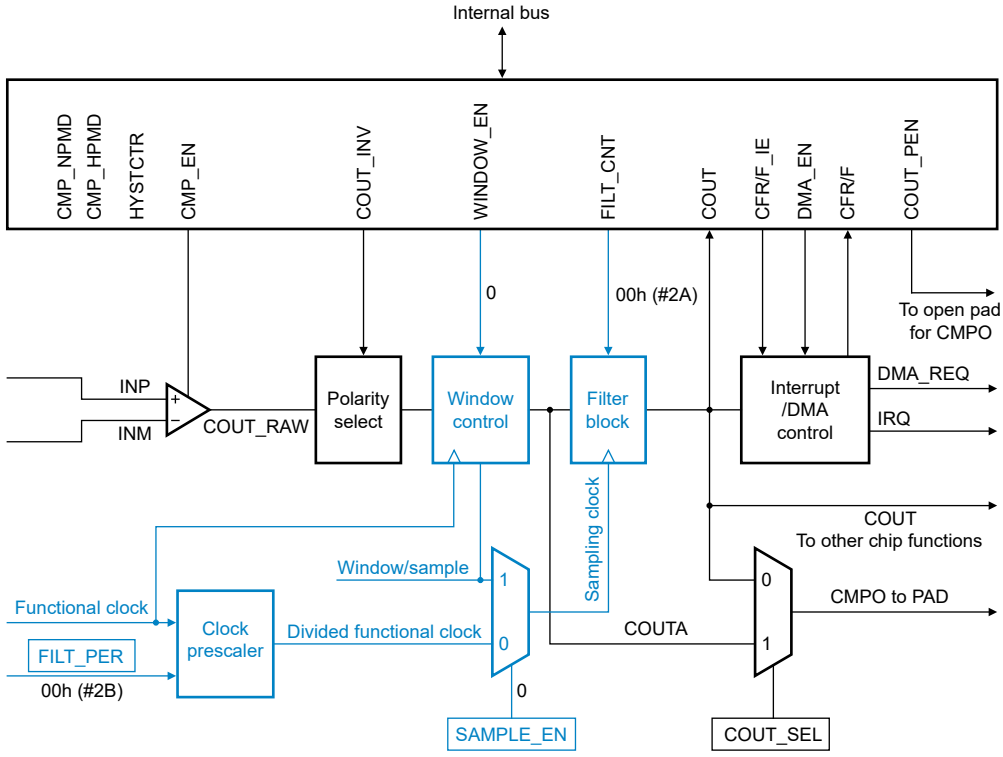


Figure 154. Comparator operation in continuous mode

COUT_RAW is optionally inverted in this mode but is not subject to external sampling or filtering. Both window control and filter blocks bypass completely, and CSR[COUT] updates continuously. The path from comparator input pins to output pins operates in a combinational (unclocked) mode. COUT and COUTA are identical in this mode.

For cases where a comparator drives a fault input, you must configure it to operate in Continuous mode so that an external fault can immediately pass to the target fault circuitry through the comparator.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

36.3.5.3 Sampled, non-filtered mode (#3A and #3B)

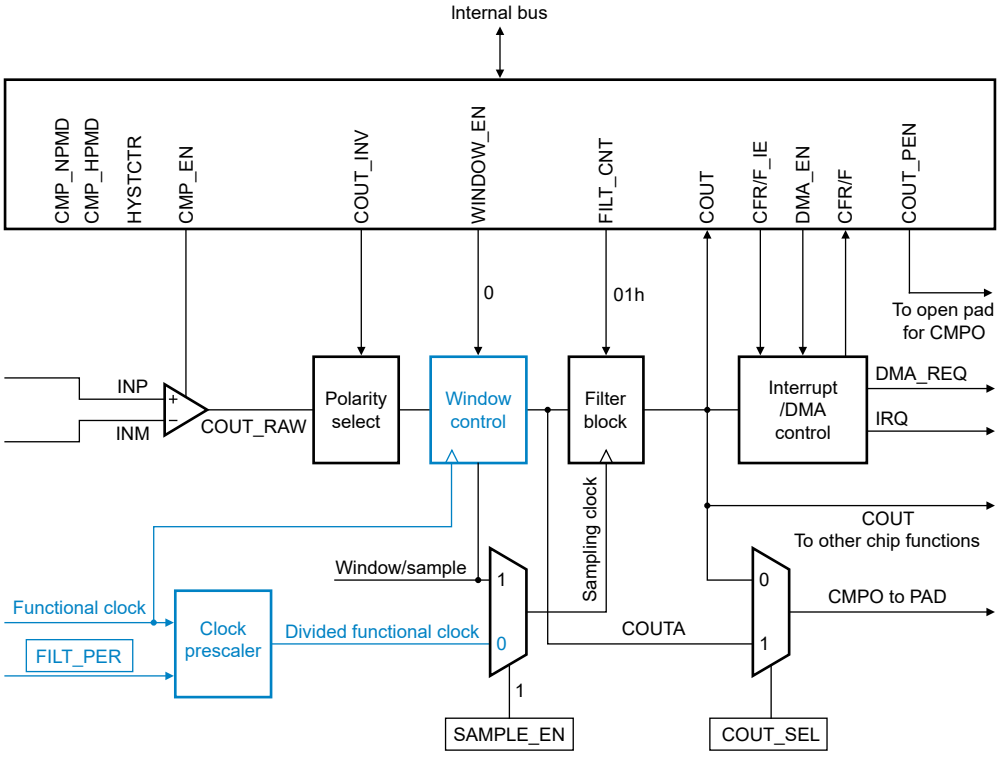


Figure 155. Sampled, non-filtered (#3A): sampling point externally derived

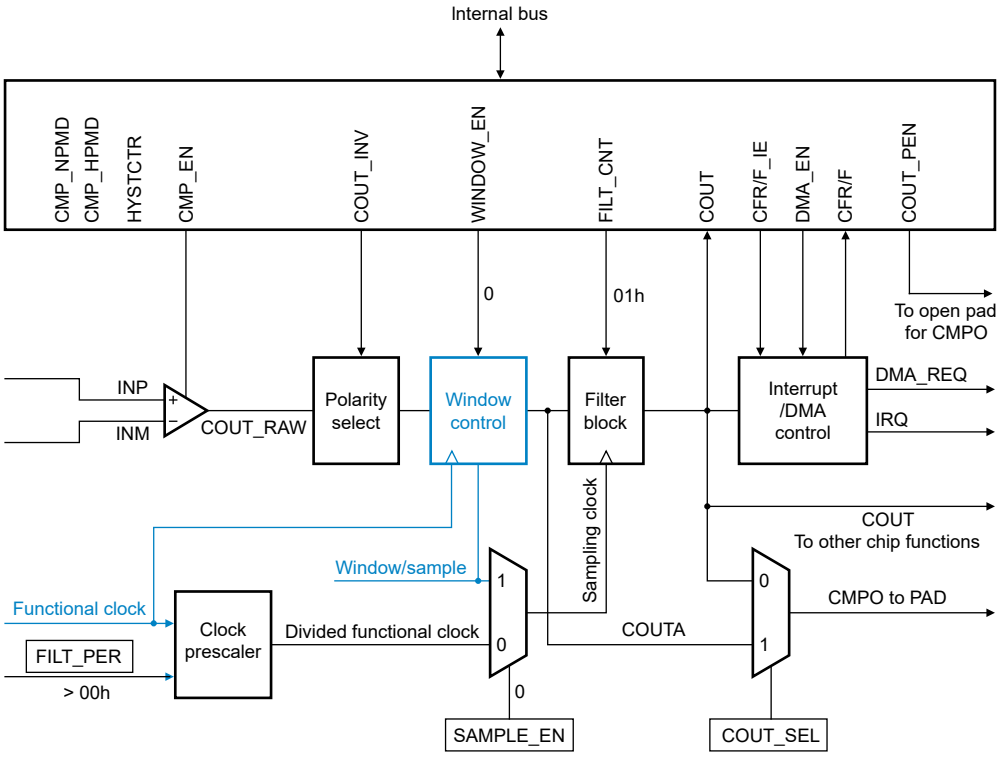


Figure 156. Sampled, Non-Filtered (#3B): sampling interval internally derived

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

In this mode, the path from analog inputs to COUTA is combinational (unclocked). Windowing control bypasses completely. You can sample COUTA whenever you detect a rising edge on the sampling clock.

The difference in two operation modes (#3A and #3B) of sampled, non-filtered mode is that how you drive the clock to the filter block. In #3A, the clock to filter block drives externally, and in #3B, the clock to filter block drives internally.

The filter block has no other function than sample or hold of the comparator output in this mode.

The following figure shows the comparator operation in this mode, assuming that the polarity select sets to a non-inverting state.

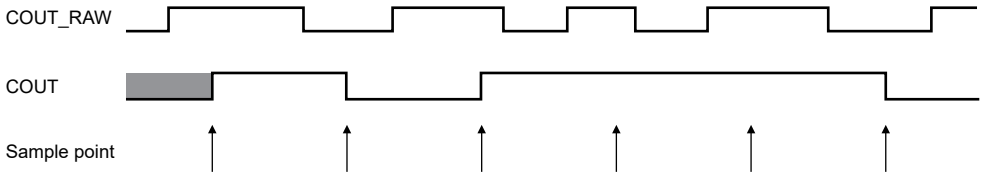


Figure 157. Sampled, Non-Filtered mode timing diagram

36.3.5.4 Sampled, filtered mode (#4A and #4B)

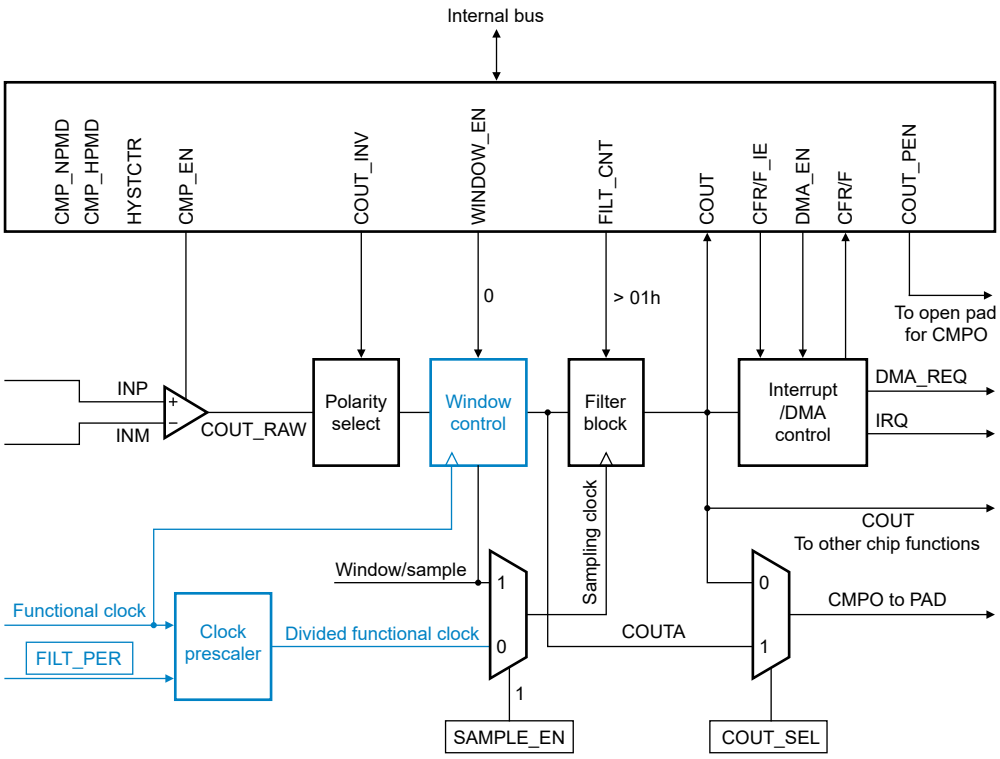


Figure 158. Sampled, filtered (#4A): sampling point externally derived

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

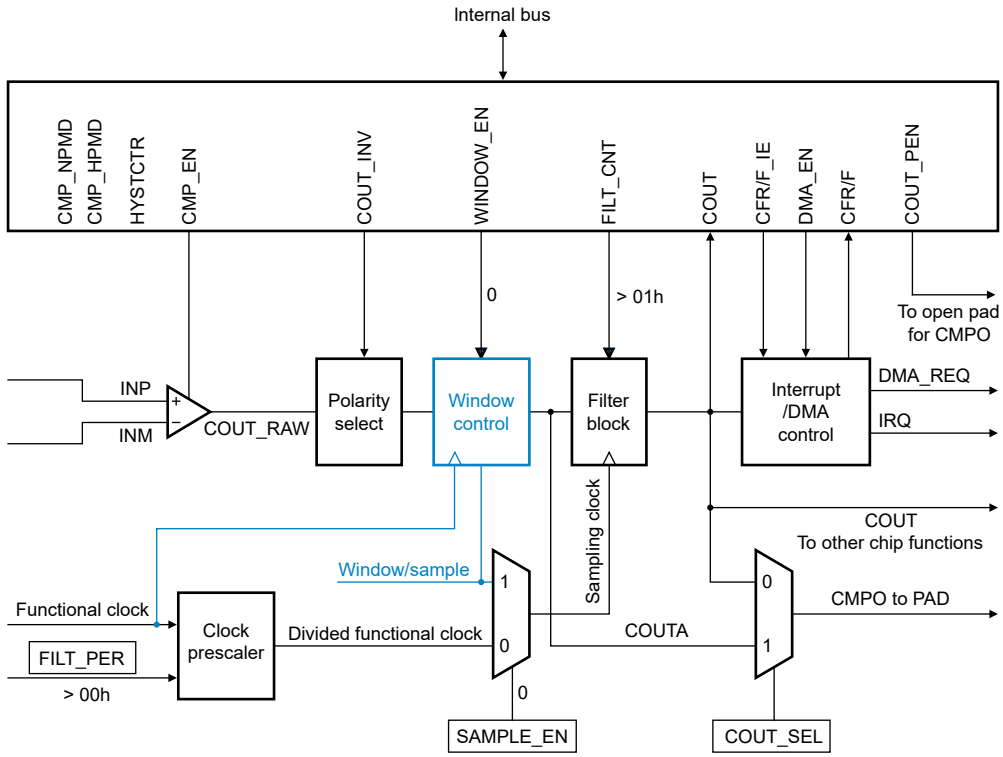


Figure 159. Sampled, filtered (#4B): sampling point internally derived

In this mode, the path from the analog inputs to COUTA is combinational(unclocked). Windowing control bypasses completely. You can sample COUTA whenever you detect a rising edge on the sampling clock.

The only difference in operation between sampled, non-filtered (#3A) mode and sampled, filtered (#4A) mode is that CCR1[FILT_CNT] is larger than 1, which activates filter operation.

The only difference in operation between sampled, non-filtered (#3B) mode and sampled, filtered (#4B) mode is that CCR1[FILT_CNT] is larger than 1, which activates filter operation.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

36.3.5.5 Windowed mode (#5A and #5B)

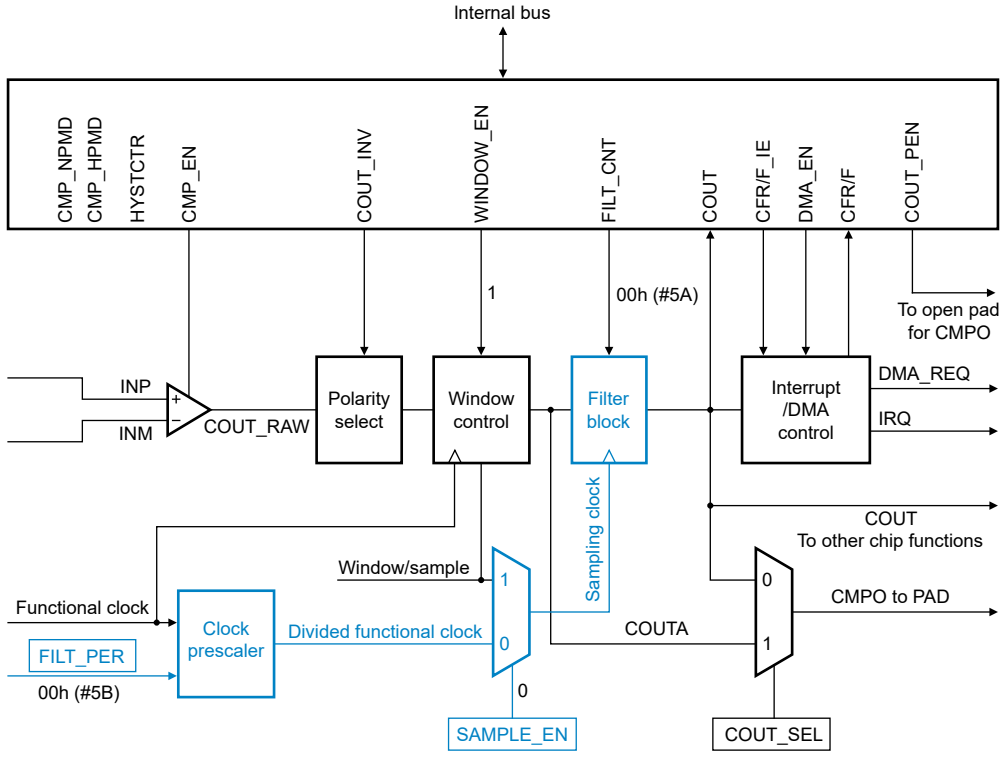


Figure 160. Windowed mode

The functional clock clocks COUTA whenever you enable the window in this mode. The last latched value holds after you disable the window and the filter block is bypassed.

The following figure shows the comparator operation in this mode, ignoring the latency of the analog comparator, polarity select, and window control block. The polarity select sets to a non-inverting state.

COUTA may lag the analog inputs by up to two functional clock cycles plus the combinational path delay through the comparator and polarity select logic in the actual operation.

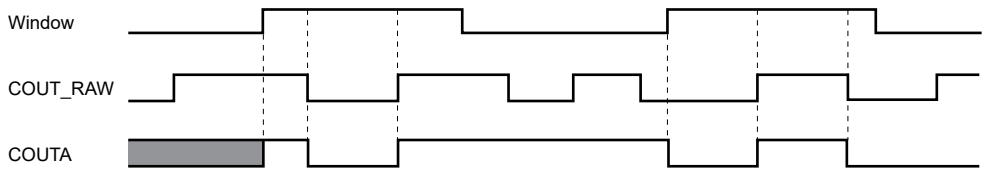


Figure 161. Windowed mode timing diagram

The following figure shows that if CCR1[COUTA_OWEN] becomes 1, you can define COUTA level as CCR1[COUTA_OW], after you closes the window.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

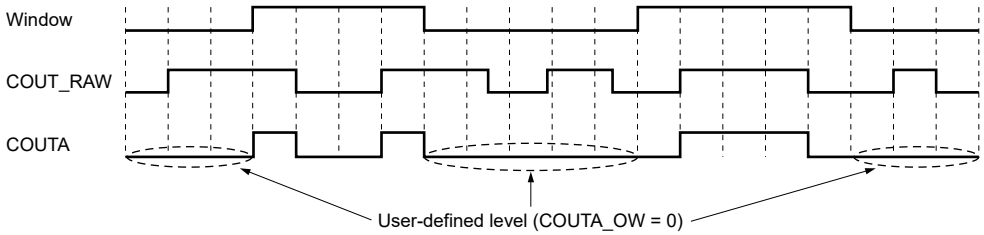


Figure 162. Windowed mode timing diagram with user defined value 0 outside window

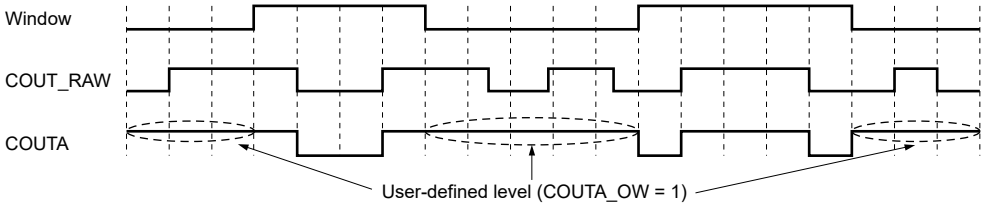


Figure 163. Windowed mode timing diagram with user defined value 1 outside window

If CCR1[WINDOW_CLS] becomes 1, you can define the COUT event (rising edge, falling edge or both edges that CCR1[EVT_SEL] selects) to close the window. The external window signal has to go to zero and back to one to enable the internal window again. The following figure shows an example that COUT rising edge closes the internal window.

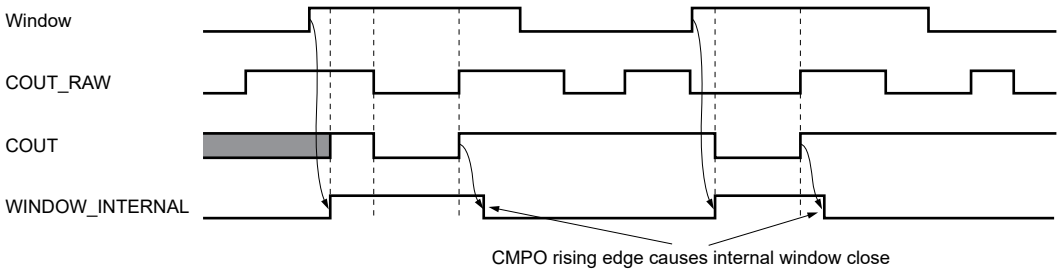


Figure 164. Windowed mode timing diagram with COUT rising edge close window

The following figure shows that if CCR1[WINDOW_INV] becomes 1, you can invert the window signal before you use it.

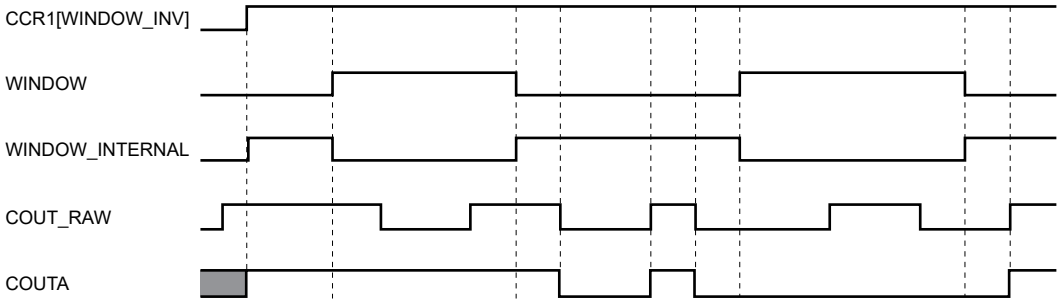


Figure 165. Windowed mode timing diagram with window signal inverted

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

36.3.5.6 Windowed/Resampled mode (#6)

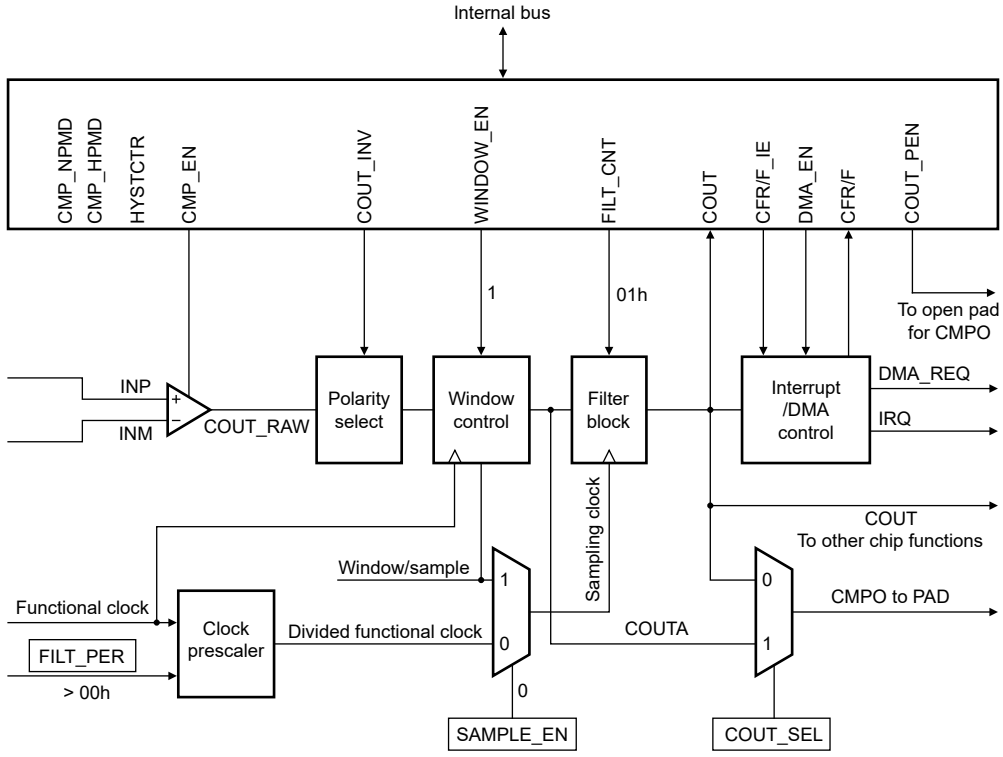


Figure 166. Windowed/Resampled mode

This mode of operation results in an unfiltered string of comparator samples where CCR1[FILT_PER] and the functional clock rate determines the interval between the samples. The following section shows that the configuration for this mode is virtually identical to that for the Windowed/Filtered mode. The only difference is that the value of CCR1[FILT_CNT] must be 1 in this mode.

The following figure uses the same input stimulus shown in Figure 161, and adds resampling of COUTA to generate COUT. The arrows in the figure indicate the time points at which the samples are taken. You can ignore prop delays and latency for clarity.

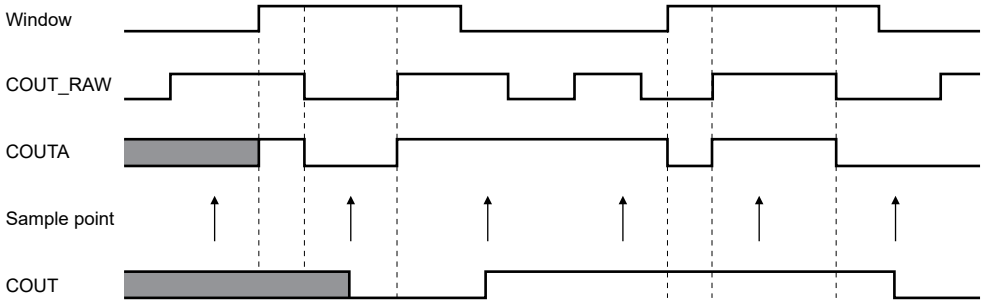


Figure 167. Windowed/Resampled mode operation

This example demonstrates the operation of the comparator in Windowed/Resampled mode, and does not reflect any specific application. Based on the sampling rate and window placement, COUT may not see zero-crossing events that the analog comparator detects. You must carefully consider the sampling period and/or window placement for a given application.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

36.3.5.7 Windowed/Filtered mode (#7)

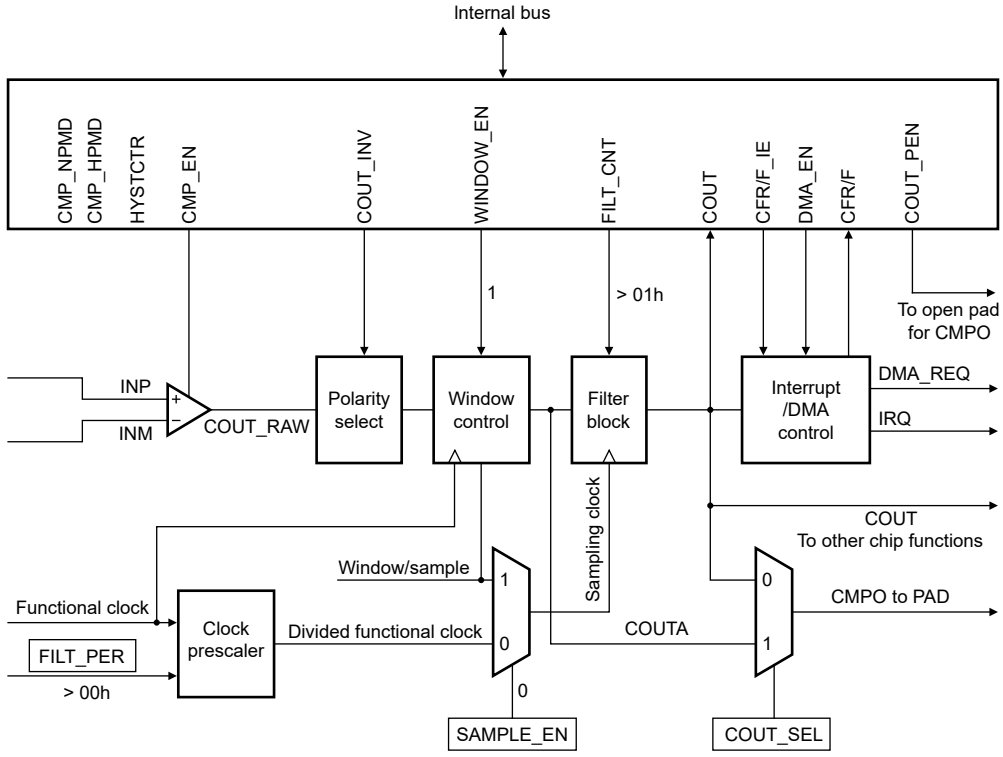


Figure 168. Windowed/Filtered mode

The only difference in operation between Windowed/Resampled mode (#6) and Windowed/Filtered mode (#7) is that CCR1[FILT_CNT] is >1, which activates filter operation.

This mode is the most complex mode of operation for the comparator block, as it utilizes both windowing and filtering features. It also has the highest latency of any of the modes. This is approximately: up to 2 peripheral clock synchronization in the window function + ((CCR1[FILT_CNT] x CCR1[FILT_PER]) + 1) x peripheral clock for the filter function.

36.3.6 DMA

After DMA is enabled by writing 1 to CCR1[DMA_EN] and interrupt is enabled by writing 1 to IER[CFR_IE], IER[CFF_IE], or both, the corresponding change on COUT forces a DMA transfer request rather than a CPU interrupt. After the DMA completes the transfer, it sends a transfer completing indicator signal that deasserts the DMA transfer request and clears the flags (both CSR[CFR] and CSR[CFF]) to allow a subsequent change on comparator output to occur and forces another DMA request.

The comparator can remain functional in Deep Sleep modes if CCR0[CMP_STOP_EN] becomes 1.

A DMA transfer request wakes up the system from Deep Sleep mode. After completing the data transfer, the system go back again to Deep Sleep mode. See the DMA chapters in this document for more information on the asynchronous DMA function.

36.3.7 Clocking

LPCMP requires the following clocks to operate:

Table 228. LPCMP clocks

Type of clock	Description
Bus	Controls the access to LPCMP registers.

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table 228. LPCMP clocks (continued)

Type of clock	Description
Functional	Controls window/filter function. LPCMP configures CCR1[FUNC_CLK_SEL] to select one from the four functional clock sources. See the Chip-specific LPCMP information for more on the functional clock source information.
Round-robin clock (RCLK)	Controls Round-robin trigger mode. LPCMP configures CCR1[RR_CLK_SEL] to select one from the four round-robin clock sources. See the Chip-specific LPCMP information for more on the round robin clock source information.

36.3.8 Resets

The global chip reset signal resets LPCMP.

36.3.9 Interrupts

After the corresponding IER becomes 1, CSR[CFR], CSR[CFF], and CSR[RRF] can generate an interrupt, assuming that CCR1[DMA_EN] is not 1. You can clear either the flag or IER to deassert the interrupt.

36.4 External signal descriptions

Below table introduces external signals.

Table 229. External signal descriptions

Signal	Description	I/O
CMPO	Filtered or unfiltered comparator output	O
Input_Analog_Channels	Analog input channels (see the chip-specific information for more on the connections).	I
VREFH_EXT	External reference voltage for the CMP-DAC (see the chip-specific information for more on the connections).	I
RR_ACTIVE	Round-robin trigger mode enabled.	O

36.5 Initialization

You can enable LPCMP by writing 1 to CCR0[CMP_EN], and then configuring the control registers (CCR1, CCR2, DCR, and so on).

To disable LPCMP, write 0 to CCR0[CMP_EN]. Switching operation modes or changing control register fields on-the-fly (when CCR0[CMP_EN] is set to 1) may cause noise on the COUT or COUTA signals. To avoid unwanted signal noise, you must ensure to disable the module before switching modes or changing control fields.

The time required to stabilize COUT is the power-on delay of the comparators plus the largest propagation delay from a selected analog source through the analog comparator, windowing function, and filter (see the Comparator and 8-bit DAC electrical specifications section of LPCMP datasheet for more information on propagation delay and power-up delay). Table 227 specifies the delay that the windowing and filter function causes.

During operation, you must always consider the propagation delay of the selected data paths. It can take many functional clock cycles for COUT and CSR[CFR]/CSR[CFF] to reflect an input change or a configuration change to one of the components involved in the data path.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

36.6 Application information

36.6.1 Round-robin trigger mode programing recommendation

Configure the Round-robin trigger mode as follows:

1. Configure RRCR0[RR_CLK_SEL] to select the RR clock source.
2. Configure the comparison cycles by RRCR0[RR_NSAM]. Note: It is a mandatory request that the round robin cycling period must set longer than the time that all the active channels complete the specified comparison cycles set by RRCR0[RR_NSAM].
3. Configure CMP initialization delay by RRCR0[RR_INITMOD]. Note: In programming RRCR0[RR_INITMOD], the RR_INITMOD x round robin clock period must be longer than the initialization delay, see the LPCMP datasheet for more information.
4. Configure RRCR0[RR_SAMPLE_CNT] and RRCR0[RR_SAMPLE_THRESHOLD].
5. Configure RRCR0[RR_TRG_SEL] to select the trigger from internal or external.
6. Enable RRCR2[RR_TIMER_EN] and configure RRCR2[RR_TIMER_RELOAD] according to the round robin clock frequency, if using an internal trigger.
7. Configure RRCR1[FIXP] to select the fixed port of CMP and RRCR1[FIXCH] to select the fixed channel.
8. Configure channels for comparison by RRCR1[RR_CHnEN].
9. Write RRCSR[RR_CHnOUT] to define the pre-set state of channel n.
10. Clear channel flags RRSR[RR_CHnF].
11. Enable round robin interrupt by IER[RRF_IE] (disable IER[CFR_IE] and IER[CFF_IE]).
12. Enable round-robin trigger mode by setting RRCR0[RR_EN] to 1.

36.6.2 Round-robin clock (RCLK) frequency requirement

(1) RCLK high frequency limit

RCLK high frequency limit depends on two facts:

1. The analog CMP and DAC initialization time (see the chip data sheet for more information on the initialization time.)
   - RRCR0[RR_INITMOD] provides a maximum 63 RCLK cycles for the analog CMP and DAC initialization.
   - RCLK must be slow to assure: $63 * (1/f_{RCLK}) > T_{initialization}$, where f_{RCLK} is in MHz, and $T_{initialization}$ is in microsecond.
   - so $f_{RCLK} < 63 / T_{initialization}$
   - Example: $T_{initialization}$ = 40 µs, then f_{RCLK} should be smaller than 1.575 MHz.
2. The analog CMP propagation delay (see the Comparator and 8-bit DAC electrical specifications section of LPCMP datasheet for more information on the CMP propagation delay.)
   - RRCR0[RR_NSAM] provides a maximum 4 RCLK cycles for the analog CMP propagation delay.
   - RCLK must be slow to assure: $4 * (1/f_{RCLK}) > T_{propagation}$, where f_{RCLK} is in MHz, and $T_{propagation}$ is in microsecond.
   - $f_{RCLK} < 4 / T_{propagation}$
   - Example: $T_{propagation}$ = 0.1 µs, then f_{RCLK} must be smaller than 40 MHz.

(2) RCLK low frequency limit

In theory, RCLK frequency has no low limit. But the lower the RCLK frequency, the longer the scan time. Therefore, the lower limit of the RCLK frequency depends on the system application.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

36.7 LPCMP register descriptions

The memory map comprises of 32-bit aligned registers, which you can access via 8-, 16- or 32-bit reads and 32-bit write. Attempted accesses using unsupported write data sizes, writes to read-only resources, or to reserved spaces terminate with an error. Read access to reserved address generates a transfer error and the read data bus shows all 0s.

36.7.1 LPCMP memory map

CMP0 base address: 400B_1000h

Offset	Register	Width (In bits)	Access	Reset value
0h	Version ID (VERID)	32	R	0100_0001h
4h	Parameter (PARAM)	32	R	0000_0002h
8h	Comparator Control Register 0 (CCR0)	32	RW	0000_0002h
Ch	Comparator Control Register 1 (CCR1)	32	RW	0000_0000h
10h	Comparator Control Register 2 (CCR2)	32	RW	0000_0000h
18h	DAC Control (DCR)	32	RW	0000_0000h
1Ch	Interrupt Enable (IER)	32	RW	0000_0000h
20h	Comparator Status (CSR)	32	RW	0000_0000h
24h	Round Robin Control Register 0 (RRCR0)	32	RW	0000_0000h
28h	Round Robin Control Register 1 (RRCR1)	32	RW	0000_0000h
2Ch	Round Robin Control and Status (RRCSR)	32	RW	0000_0000h
30h	Round Robin Status (RRSR)	32	RW	0000_0000h
38h	Round Robin Control Register 2 (RRCR2)	32	RW	0000_0000h

36.7.2 Version ID (VERID)

Offset

Register	Offset
VERID	0h

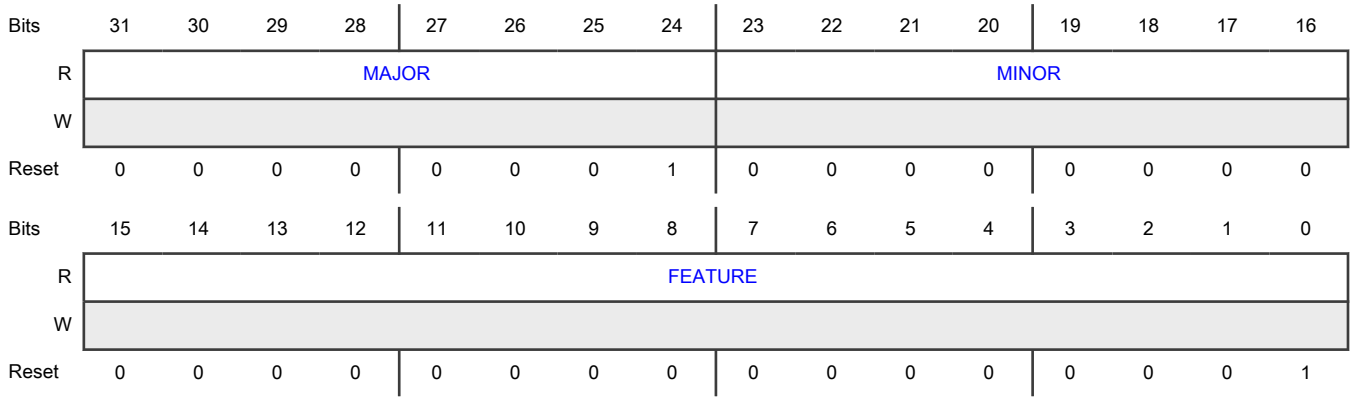
Function

Contains version numbers for the module design and feature set.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram



Fields

Field	Function
31-24 MAJOR	Major Version Number Returns the major version number for the module design.
23-16 MINOR	Minor Version Number Returns the minor version number for the module design.
15-0 FEATURE	Feature Specification Number Returns the feature set number. 0000_0000_0000_0001b - Round robin feature

36.7.3 Parameter (PARAM)

Offset

Register	Offset
PARAM	4h

Function

Contains parameter values that are implemented in the module.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved												DAC_RES			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

Fields

Field	Function
31-4 —	Reserved
3-0 DAC_RES	DAC Resolution Indicates supported DAC resolutions. **NOTE** All other bit field values are reserved. 0000b - 4-bit DAC 0001b - 6-bit DAC 0010b - 8-bit DAC 0011b - 10-bit DAC 0100b - 12-bit DAC 0101b - 14-bit DAC 0110b - 16-bit DAC

36.7.4 Comparator Control Register 0 (CCR0)

Offset

Register	Offset
CCR0	8h

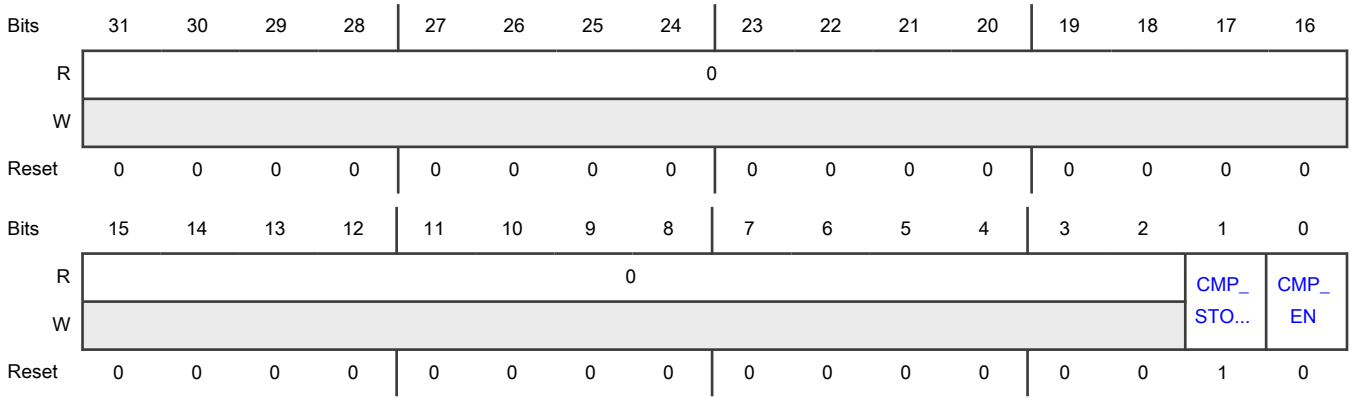
Function

Contains configuration options for enabling the analog comparator.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram



Fields

Field	Function
31-2 —	Reserved
1 CMP_STOP_EN	Comparator Deep Sleep Mode Enable Enables the analog comparator or the DAC when the module is in Deep Sleep mode. **NOTE** This field has no effect in Round-robin Trigger mode. 0b - Disables the analog comparator regardless of CMP_EN. 1b - Allows CMP_EN to enable the analog comparator.
0 CMP_EN	Comparator Enable Enables the analog comparator. 0b - Disable (The analog logic remains off and consumes no power.) 1b - Enable

36.7.5 Comparator Control Register 1 (CCR1)

Offset

Register	Offset
CCR1	Ch

Function

Contains configuration options for the comparator operation, such as enabling Sampling or Windowing mode.

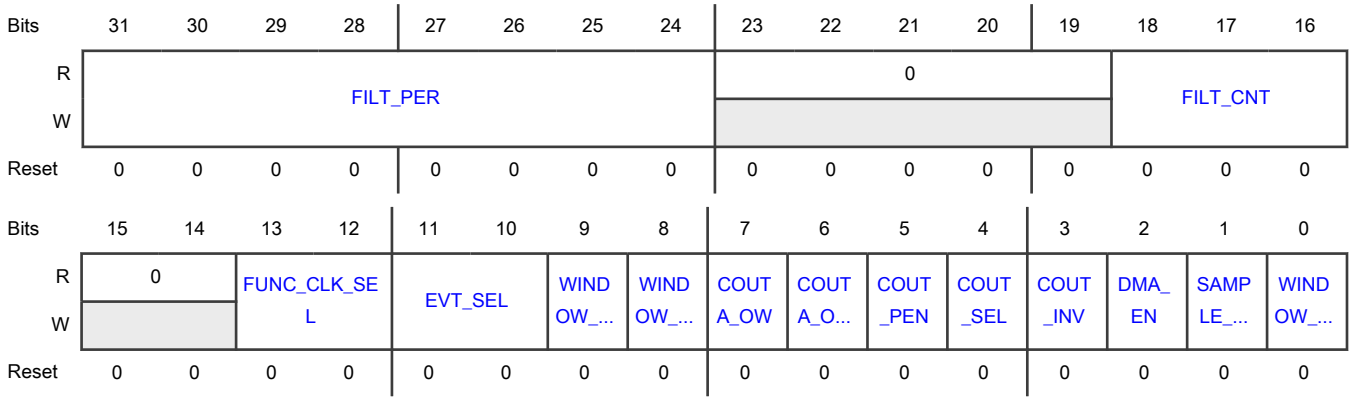
NOTE

You cannot enable Sampling and Windowing modes both at the same time. Sampling mode takes precedence over Windowing mode. If you write 1 to both SAMPLE_EN and WINDOW_EN, only SAMPLE_EN becomes 1.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram



Fields

Field	Function
31-24 FILT_PER	Filter Sample Period Specifies the sampling period (in functional clock cycles) of the comparator output filter. Programming this field to 00h bypasses the filter. See Functional description for more information on filter programming and latency. **NOTE** FILT_PER has no effect in Sampling mode (CCR1[SAMPLE_EN] = 1).
23-19 —	Reserved
18-16 FILT_CNT	Filter Sample Count Specifies the number of consecutive samples that must agree before the comparator output filter accepts the sample as a new valid output state. See Functional description for more information on filter programming and latency. 000b - Filter is bypassed: COUT = COUTA 001b - 1 consecutive sample (Comparator output is simply sampled.) 010b - 2 consecutive samples 011b - 3 consecutive samples 100b - 4 consecutive samples 101b - 5 consecutive samples 110b - 6 consecutive samples 111b - 7 consecutive samples
15-14 —	Reserved
13-12	Functional Clock Source Select

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
FUNC_CLK_SEL	Selects which clock source is used for the functional clock. See the Chip-specific LPCMP information for more on the functional clock source information. 00b - Select functional clock source 0 01b - Select functional clock source 1 10b - Select functional clock source 2 11b - Select functional clock source 3
11-10 EVT_SEL	COUT Event Select Selects which COUT signal edge (rising, falling, or both) defines a COUT event. **NOTE** Valid only in Windowing mode. 00b - Rising edge 01b - Falling edge 1xb - Both edges
9 WINDOW_CLS	COUT Event Window Close Enables a COUT event (defined as a COUT rising edge, falling edge, or both) to close an active window. See EVT_SEL to configure the COUT event. **NOTE** The WINDOW signal has to go to zero and back to one again to re-activate the window. Valid only in Windowing mode. 0b - COUT event cannot close the window 1b - COUT event can close the window
8 WINDOW_INV	WINDOW/SAMPLE Signal Invert Inverts the window/sample signal. 0b - Do not invert 1b - Invert
7 COUTA_OW	COUTA Output Level for Closed Window Defines the COUTA signal value when the window is closed. **NOTE** Valid only in Windowing mode and when COUTA_OWEN=1. 0b - COUTA is 0 1b - COUTA is 1
6	COUTA_OW Enable

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
COUTA_OWEN	Enables the COUTA signal value to be defined by COUTA_OW when the window is closed. **NOTE** Valid only in Windowing mode. 0b - COUTA holds the last sampled value. 1b - Enables the COUTA signal value to be defined by COUTA_OW.
5 COUT_PEN	Comparator Output Pin Enable Enables the comparator output to become an available signal option for a selected package pin. 0b - Not available 1b - Available
4 COUT_SEL	Comparator Output Select Selects which comparator output option, COUT or COUTA, to use for CMPO. 0b - Use COUT (filtered) 1b - Use COUTA (unfiltered)
3 COUT_INV	Comparator Invert Selects the polarity of the analog comparator function, affecting the value driven to the COUT output (on both the chip pin and as CSR[COUT]) when CCR0[CMP_EN] is 0. **NOTE** COUT_INV has no effect in Trigger mode. 0b - Do not invert 1b - Invert
2 DMA_EN	DMA Enable Enables DMA transfers triggered from the LPCMP module. After this field and the corresponding interrupt enable field becomes 1, a DMA request is asserted when CFR or CFF becomes 1. 0b - Disable 1b - Enable
1 SAMPLE_EN	Sampling Enable Enables Sampling mode. 0b - Disable 1b - Enable
0 WINDOW_EN	Windowing Enable Enables Windowing mode.

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	NOTE Valid only when SAMPLE_EN = 0. 0b - Disable 1b - Enable

36.7.6 Comparator Control Register 2 (CCR2)

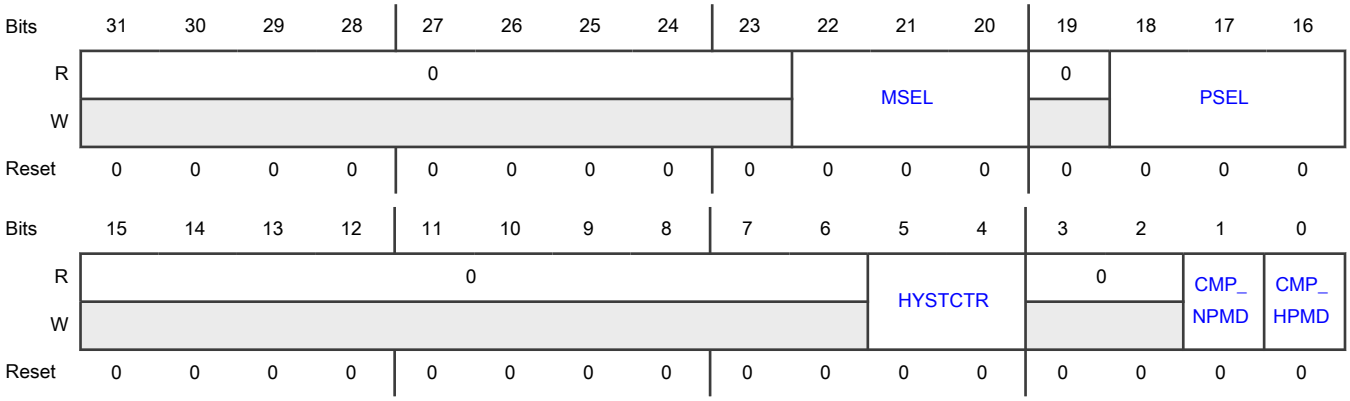
Offset

Register	Offset
CCR2	10h

Function

Contains the configuration options for the comparator operation, such as selecting the plus and minus comparator inputs and the hysteresis levels.

Diagram



Fields

Field	Function
31-23 —	Reserved
22-20 MSEL	Minus Input MUX Select Selects the input used for the negative mux. See the chip-specific LPCMP information for more on connections.

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	NOTE MSEL has no effect in Trigger mode. 000b - Input 0m 001b - Input 1m 010b - Input 2m 011b - Input 3m 100b - Input 4m 101b - Input 5m 110b - Reserved 111b - Internal DAC output
19 —	Reserved
18-16 PSEL	Plus Input MUX Select Selects the input used for the positive mux. See the chip-specific LPCMP information for more on connections. **NOTE** PSEL has no effect in Trigger mode. 000b - Input 0p 001b - Input 1p 010b - Input 2p 011b - Input 3p 100b - Input 4p 101b - Input 5p 110b - Reserved 111b - Internal DAC output
15-6 —	Reserved
5-4 HYSTCTR	Comparator Hysteresis Control Selects the level of internally generated hysteresis for the comparator output. **NOTE** This applies to the comparator hard block.

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	00b - Level 0: Analog comparator hysteresis 0 mV. 01b - Level 1: Analog comparator hysteresis 10 mV. 10b - Level 2: Analog comparator hysteresis 20 mV. 11b - Level 3: Analog comparator hysteresis 30 mV.
3-2 —	Reserved
1 CMP_NPMD	CMP Nano Power Mode Select Enables Nano Power mode for the comparator. **NOTE** If the LPCMP is used in deep sleep (or lower) power modes, LPCMP need to enable the nano-power mode via LPCMP_CRR2_CMP_NPMD. 0b - Disables CMP Nano power mode. CCR2[CMP_HPMD] determines the mode for the comparator. 1b - Enables CMP Nano power mode.
0 CMP_HPMD	CMP High Power Mode Select Selects Low or High Power(Speed) mode for the comparator. **NOTE** Valid only when not in Nano Power mode (CMP_NPMD = 0). 0b - Low power (speed) comparison mode 1b - High power (speed) comparison mode

36.7.7 DAC Control (DCR)

Offset

Register	Offset
DCR	18h

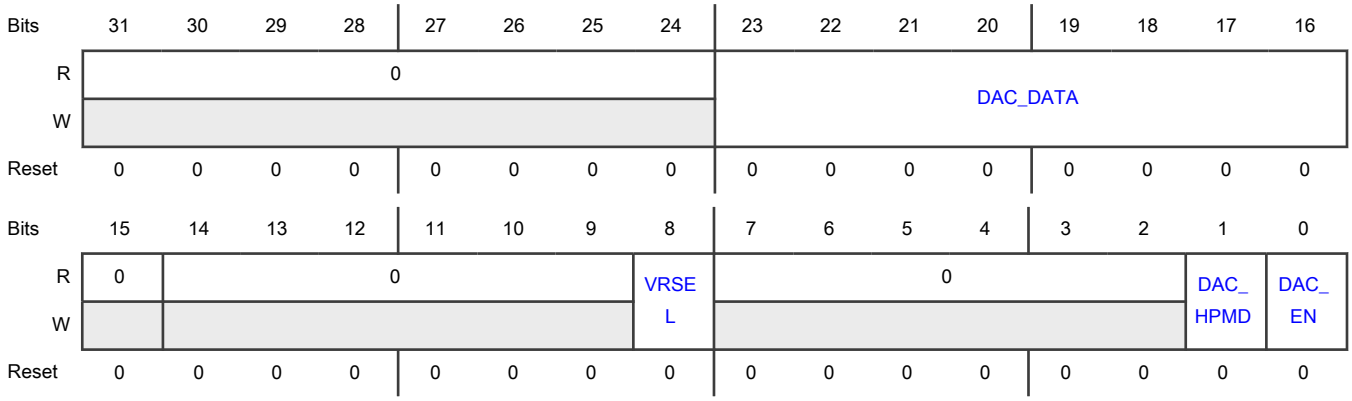
Function

Contains the configuration options to enable the DAC.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram



Fields

Field	Function
31-24 —	Reserved
23-16 DAC_DATA	DAC Output Voltage Select Selects the DAC output (DACO) voltage from one of 256 distinct levels by configuring the value of DAC_DATA. The DACO ranges from Vin/256 to Vin. **NOTE** DACO = (Vin/256) * (DAC_DATA + 1)
15 —	Reserved
14-9 —	Reserved
8 VRSEL	DAC Reference High Voltage Source Select Selects the high voltage reference source for the Vin supply of the DAC's resistor ladder network. See the chip-specific LPCMP information for the source of vrefh0 and vrefh1. 0b - VREFH0 1b - VREFH1
7-2 —	Reserved
1 DAC_HPMD	DAC High Power Mode Enables the DAC high power mode. 0b - Disable 1b - Enable

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
0 DAC_EN	DAC Enable Enables the DAC. When disabled, power-down the DAC to conserve power. 0b - Disable 1b - Enable

36.7.8 Interrupt Enable (IER)

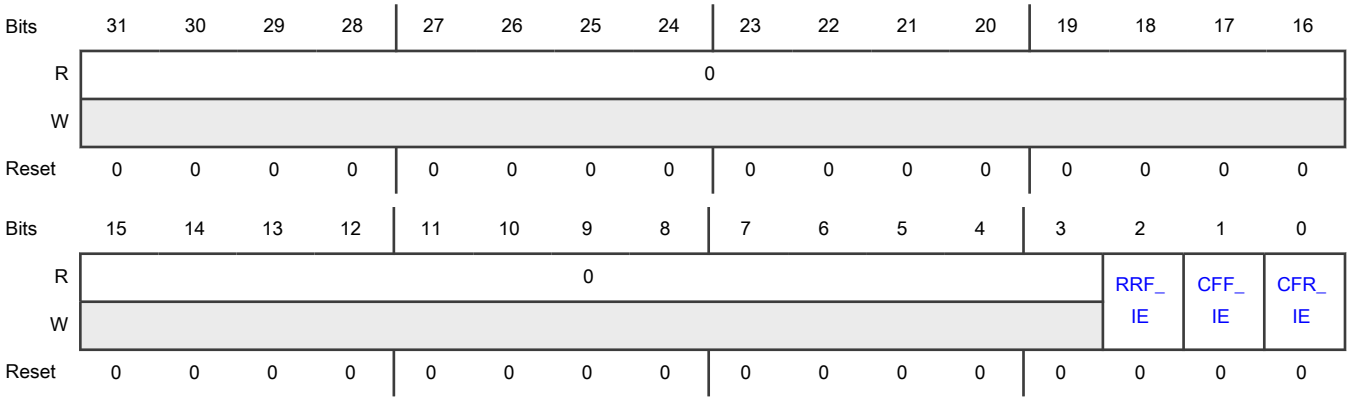
Offset

Register	Offset
IER	1Ch

Function

Provides enable fields for the comparator and round-robin flag interrupts.

Diagram



Fields

Field	Function
31-3 —	Reserved
2 RRF_IE	Round-Robin Flag Interrupt Enable Enables or disables the round-robin flag interrupt. 0b - Disables the round-robin flag interrupt. 1b - Enables the round-robin flag interrupt when the comparison result changes for a given channel.

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
1 CFF_IE	Comparator Flag Falling Interrupt Enable Enables or disables the comparator flag falling interrupt. 0b - Disables the comparator flag falling interrupt. 1b - Enables the comparator flag falling interrupt when CFF is set.
0 CFR_IE	Comparator Flag Rising Interrupt Enable Enables or disables the comparator flag rising interrupt. 0b - Disables the comparator flag rising interrupt. 1b - Enables the comparator flag rising interrupt when CFR is set.

36.7.9 Comparator Status (CSR)

Offset

Register	Offset
CSR	20h

Function

Indicates comparator status, including COUT, CFF, CFR, and RRF.

NOTE

LPCMP may output a glitch and affect the value of CSR[CFF] and CSR[CFR] at the moment of enabling CMP. In order to ensure correctness, it is recommended to write one to clear (W1C) CSR[CFF] and CSR[CFR] before further configuring CMP.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0							COUT	0					RRF	CFF	CFR
W														W1C	W1C	W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Fields

Field	Function
31-9 —	Reserved
8 COUT	Analog Comparator Output Returns the current value of the analog comparator output when read. This field resets to 0 and reads as CCR1[COUT_INV] after the analog comparator module disables when CCR0[CMP_EN] = 0. Writing to this field is ignored.
7-3 —	Reserved
2 RRF	Round-Robin Flag Detects when any channel's last comparison result is different from the pre-set value in Trigger mode. Write 1 to clear this field. This field clears when CCR0[CMP_EN] or RRCR0[RR_EN] is not 1. 0b - Not detected 1b - Detected
1 CFF	Analog Comparator Flag Falling Detects when a falling edge on COUT occurs. Write 1 to clear this field when CCR1[DMA_EN] is disabled. If CCR1[DMA_EN] is enabled, the flag automatically clears after DMA is done. This field clears when CCR0[CMP_EN] is not 1. 0b - Not detected 1b - Detected
0 CFR	Analog Comparator Flag Rising Detects when a rising edge on COUT occurs. Write 1 to clear this field when CCR1[DMA_EN] is disabled. If CCR1[DMA_EN] is enabled, the flag automatically clears after DMA is done. This field clears when CCR0[CMP_EN] is not 1. 0b - Not detected 1b - Detected

36.7.10 Round Robin Control Register 0 (RRCR0)

Offset

Register	Offset
RRCR0	24h

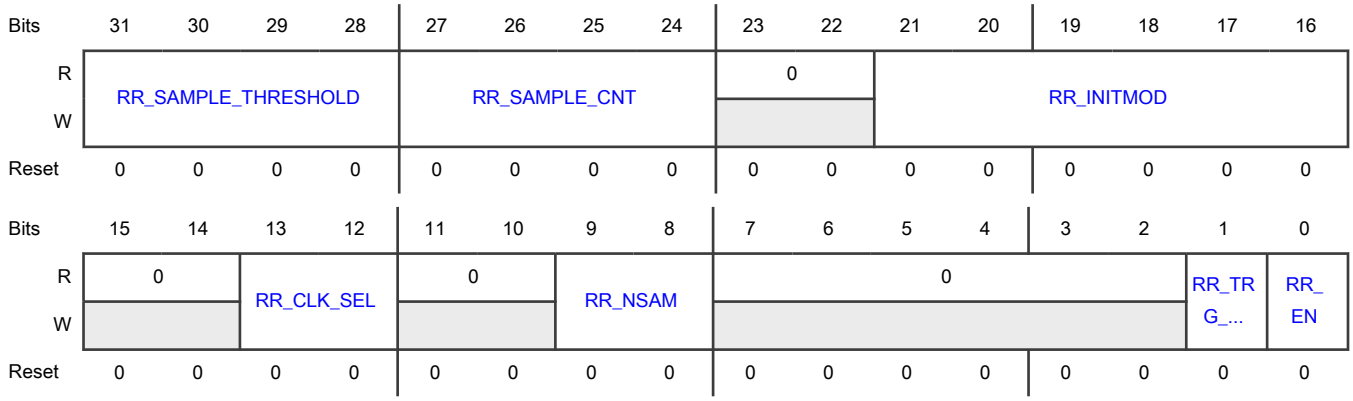
Function

Contains configuration options for the round-robin operation, such as enabling it and specifying the initialization delay.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram



Fields

Field	Function
31-28 RR_SAMPLE_THRESHOLD	Sample Time Threshold Specifies that for one channel, when (RR_SAMPLE_THRESHOLD+1) sample results are "1", the final result is "1"; otherwise the final result is "0". **NOTE** This field must not be larger than RRCR0[RR_SAMPLE_CNT]. 0000b - At least 1 sampled "1", the final result is "1" 0001b - At least 2 sampled "1", the final result is "1" 0010b - At least 3 sampled "1", the final result is "1" 0011b - At least 4 sampled "1", the final result is "1" 0100b - At least 5 sampled "1", the final result is "1" 0101b - At least 6 sampled "1", the final result is "1" 0110b - At least 7 sampled "1", the final result is "1" 0111b - At least 8 sampled "1", the final result is "1" 1000b - At least 9 sampled "1", the final result is "1" 1001b - At least 10 sampled "1", the final result is "1" 1010b - At least 11 sampled "1", the final result is "1" 1011b - At least 12 sampled "1", the final result is "1" 1100b - At least 13 sampled "1", the final result is "1" 1101b - At least 14 sampled "1", the final result is "1" 1110b - At least 15 sampled "1", the final result is "1" 1111b - At least 16 sampled "1", the final result is "1"
27-24	Number of Sample for One Channel Specifies the number of samples for one channel.

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
RR_SAMPLE_CNT	0000b - 1 samples 0001b - 2 samples 0010b - 3 samples 0011b - 4 samples 0100b - 5 samples 0101b - 6 samples 0110b - 7 samples 0111b - 8 samples 1000b - 9 samples 1001b - 10 samples 1010b - 11 samples 1011b - 12 samples 1100b - 13 samples 1101b - 14 samples 1110b - 15 samples 1111b - 16 samples
23-22 —	Reserved
21-16 RR_INITMOD	Initialization Delay Modulus Specifies the number of round-robin clock cycles that determines the comparator and DAC initialization delay specified in the chip datasheet. Calculate the initialization delay as RR_INITMOD * (round-robin clock period). For example, if the initialization delay is 80us and the round-robin clock is 100kHz, program RR_INITMOD to be 80us/10us = 8. 00_0000b - 63 cycles (same as 111111b) 00_0001b-11_1111b - 1 to 63 cycles
15-14 —	Reserved
13-12 RR_CLK_SEL	Round Robin Clock Source Select Selects which clock source is used for the Round Robin clock. See the chip-specific LPCMP information for more on the Round Robin clock source information. 00b - Select Round Robin clock Source 0 01b - Select Round Robin clock Source 1

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	10b - Select Round Robin clock Source 2 11b - Select Round Robin clock Source 3
11-10 —	Reserved
9-8 RR_NSAM	Number of Sample Clocks Specifies the number of the round-robin clock cycles to wait after scanning the active channel before sampling the channel's comparison result. After the next cycle of the round-robin clock, the sampling takes place RR_NSAM clocks later. 00b - 0 clock 01b - 1 clock 10b - 2 clocks 11b - 3 clocks
7-2 —	Reserved
1 RR_TRG_SEL	Round-Robin Trigger Select Selects the internal trigger or external trigger as the trigger source. 0b - External trigger 1b - Internal trigger
0 RR_EN	Round-Robin Enable Enables the round-robin operation. 0b - Disable 1b - Enable

36.7.11 Round Robin Control Register 1 (RRCR1)

Offset

Register	Offset
RRCR1	28h

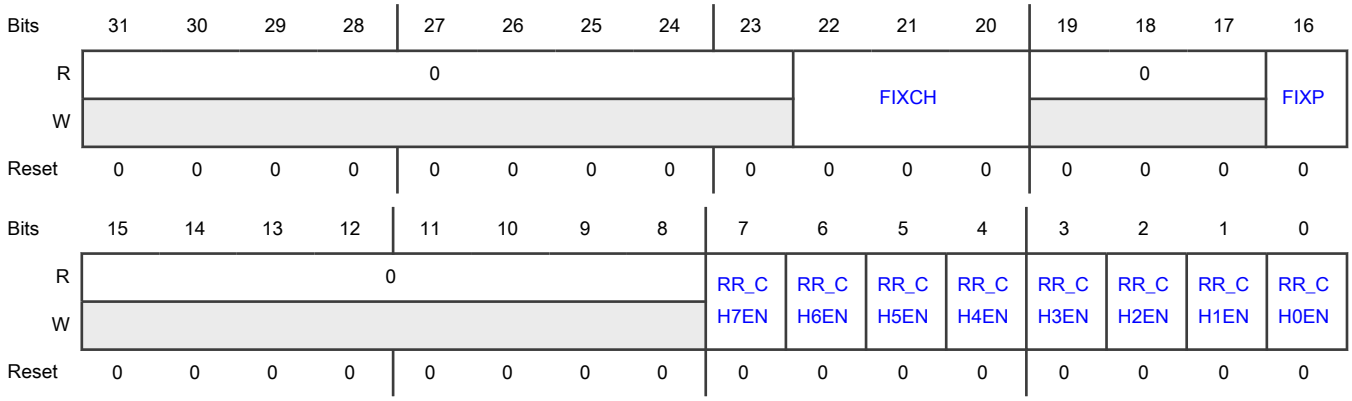
Function

Contains configuration options for the round-robin operation, such as enabling individual channels to participate.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram



Fields

Field	Function
31-23 —	Reserved
22-20 FIXCH	Fixed Channel Select Selects which channel in the mux port to fix for a given round-robin trigger mode application. 000b - Channel 0 001b - Channel 1 010b - Channel 2 011b - Channel 3 100b - Channel 4 101b - Channel 5 110b - Channel 6 111b - Channel 7
19-17 —	Reserved
16 FIXP	Fixed Port Fixes an analog mux port (plus or minus) for round-robin trigger mode. The inputs to the non-fixed port sweep during each round. 0b - Fix the plus port. Sweep only the inputs to the minus port. 1b - Fix the minus port. Sweep only the inputs to the plus port.
15-8 —	Reserved
7-0	Channel n Input Enable in Trigger Mode

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
RR_CHnEN	Enables channel n of the non-fixed mux port to check its voltage value when in Trigger mode. 0b - Disable 1b - Enable

36.7.12 Round Robin Control and Status (RRCSR)

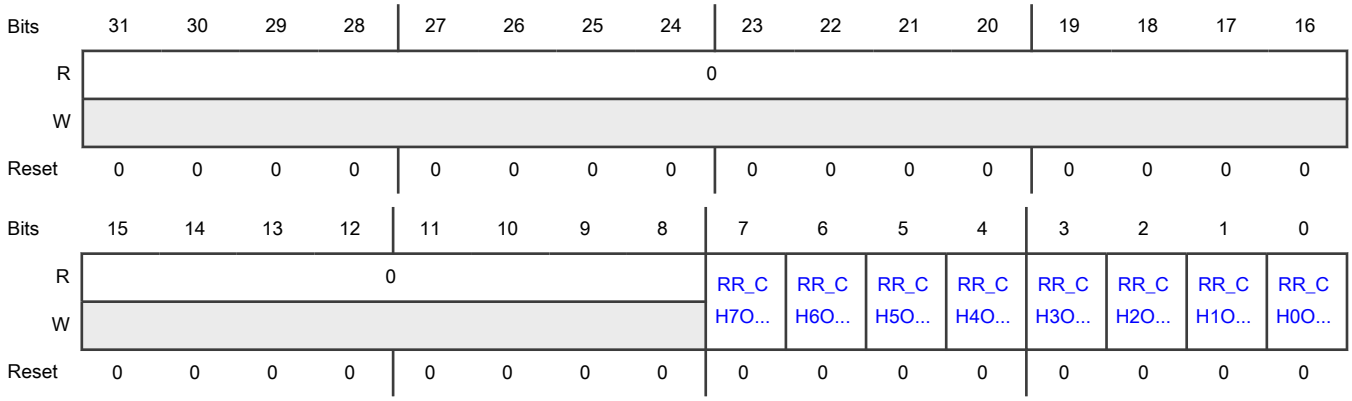
Offset

Register	Offset
RRCSR	2Ch

Function

Contains the latest comparison results of the individual channels with the fixed mux port. It also allows you to define the pre-set state for each channel.

Diagram



Fields

Field	Function
31-8 —	Reserved
7-0 RR_CHnOUT	Comparison Result for Channel n Returns the latest comparison result for channel n when read and defines the pre-set state for channel n when written to.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

36.7.13 Round Robin Status (RRSR)

Offset

Register	Offset
RRSR	30h

Function

Contains individual channel flags that indicates when a channel's last comparison result is different from its pre-set value.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								RR_CH7F	RR_CH6F	RR_CH5F	RR_CH4F	RR_CH3F	RR_CH2F	RR_CH1F	RR_CH0F
W									W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-8 —	Reserved
7-0 RR_CHnF	Channel n Input Changed Flag Indicates when the corresponding channel's last comparison result is different from its pre-set value. **NOTE** To clear a flag, write a 1 to it. 0b - No different 1b - Different

36.7.14 Round Robin Control Register 2 (RRCR2)

Offset

Register	Offset
RRCR2	38h

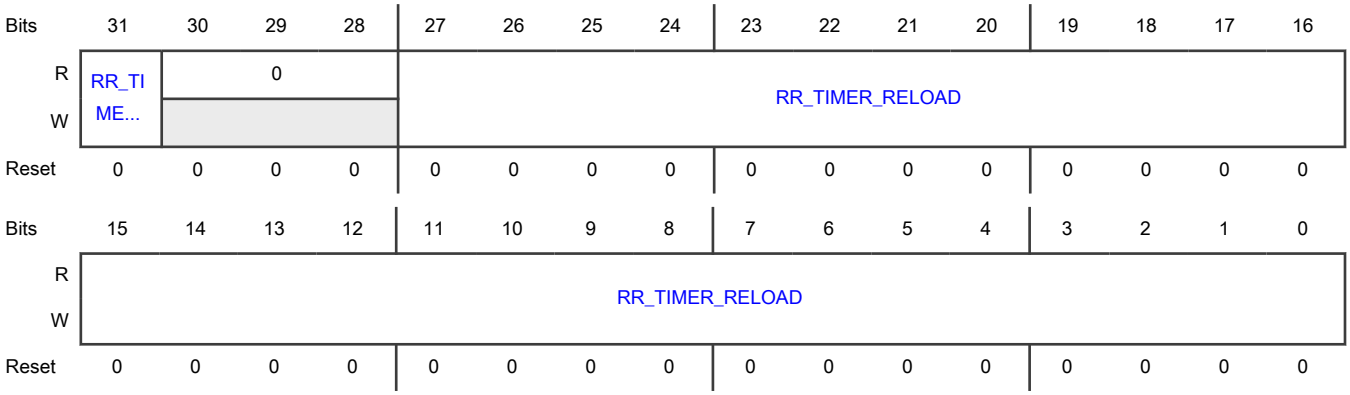
All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Function

Provides the controls for the round-robin internal trigger generation.

Diagram



Fields

Field	Function
31 RR_TIMER_EN	Round-Robin Internal Timer Enable Enables round-robin internal timer. 0b - Disables 1b - Enables
30-28 —	Reserved
27-0 RR_TIMER_RELOAD	Number of Sample Clocks Establishes the repetitive count rate for the round-robin internal timer. Each time the timer counts zero it is reloaded with this value. The round-robin trigger signal generates at a rate of (RR_TIMER_RELOAD + 1) times the round-robin clock period.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Chapter 37
Debug

37.1 Introduction

Cortex-M23 debug functionality is based on ARM CoreSight debug infrastructure, and includes processor halt, single-step, processor core register access, Vector Catch, unlimited software breakpoints, and full system memory access. The processor also includes support for hardware breakpoints and watchpoints configured during implementation:

- A breakpoint unit supporting two instruction comparators.
- A watchpoint unit supporting two watchpoint comparators.

The Cortex-M23 processor supports system level debug authentication to control access from a debugger to resources and memory.

Features:

- Arm CoreSight Architecture adapted for software trace and debug.
- Serial wire debug (SWD) and Debug Access Port (DAP) to support debugger tools.
- The following are supported for the Cortex-M23 subsystem:
  - Watchpoint Unit
  - Breakpoint Unit (BPU)
- SWD is the debug interface of the chip.

The following figures show the system level debug architecture.

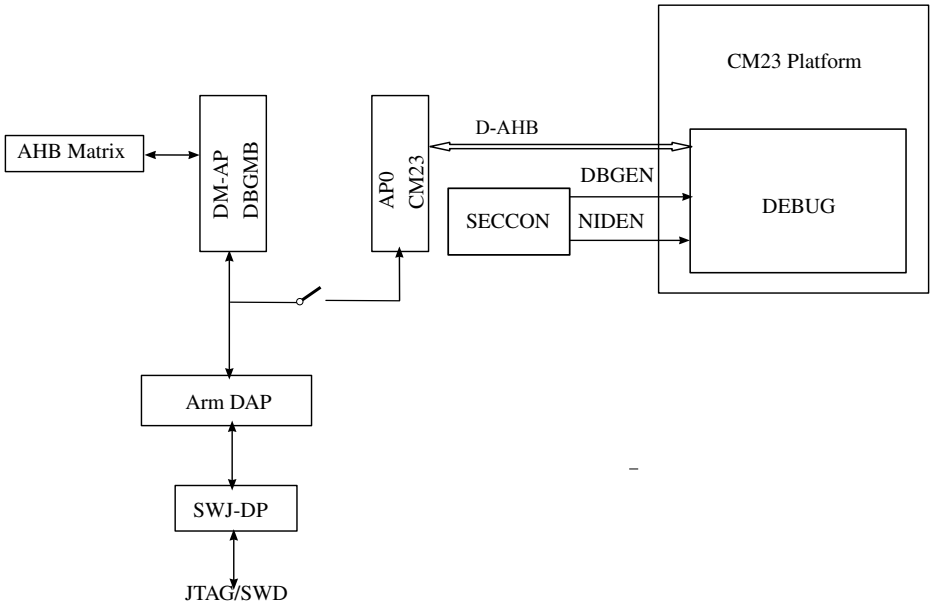


Figure 169. Debug architecture

All information provided in this document is subject to legal disclaimers.
© 2026 NXP B.V. All rights reserved.

37.2 Debug component

37.2.1 D-AHB interface

The 32-bit Debug AHB (D-AHB) interface implements the AMBA 5 AHB protocol. It can be used with a CoreSight AHB-AP to provide debugger access to all processor control and debug resources, and a view of memory that is consistent with that observed by load/store instructions acting on the processor.

37.2.2 AHB-AP debug access port

The AHB-AP is an optional debug access port into the processor system that provides access to all memory and registers in the system, including processor registers through the SCS. System access is independent of the processor status. SWJ-DP is used to access the AHB-AP.

37.2.3 DBGMB - access port (DM-AP)

DM-AP is the debug access port for DBGMB. DBGMB is used to support password based debug authentication on this chip. This port is always enabled.

37.2.4 Cortex M23 debug

This section covers the debug mechanisms associated with the CM23 processor.

37.2.4.1 Breakpoint Unit

The CM23 Breakpoint Unit (BPU) module comprises up to 2 instruction address comparators. It provides breakpoint functionality on all instructions fetched across the entire address range in which code can be located.

37.3 Debug port connectivity

The following figure shows the connectivity of debug port on this device.

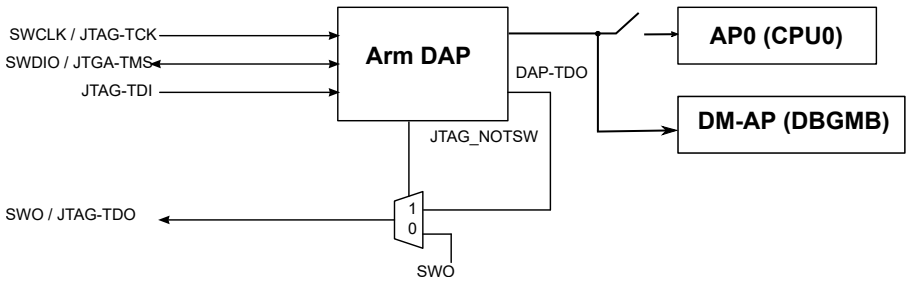


Figure 170. Debug port connectivity

The JTAG and SWD signals are summarized in the following table.

Table 230. JTAG/SWD signals summary

Pin Name	JTAG		SWD		Internal pull up/down
	Type	Description	Type	Description	
TMS/SWDIO	I/O	Mode selection	I/O	Data	Pull-up
TCLK/SWCLK/ TRACECLKIN	I	Clock	I	Clock	Pull-down
TDI	I	Data input	--	--	Pull-up
TDO	O	Data output	--	--	N/A

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

JTAG mode is the default connection of the JTAG/SWD port. The following sequence of events would switch the debug port to SWD mode.

JTAG-to-SWD change sequence has following steps:

- Send more than 50 TCLK cycles with TMS(SWDIO) = 1
- Send the 16-bit sequence on TMS(SWDIO) = 0111_1001_1110_0111 (MSB transmitted first)
- Send more than 50 TCLK cycles with TMS(SWDIO) = 1

Following is the JTAG ID table for this device.

JTAG ID	Version	Part number	Manufacture	mandatory
Value	0000b	0111 0010 0111b	000 0001 0101b	1b

37.3.1 Joint Test Access Group (JTAG)

37.3.1.1 JTAG instruction registers

The JTAG TEST TAP is connected in parallel with Arm DAP controller, that is, the JTAG-DP port of the DAP. The Instruction Register (IR) length is 4-bits. The JTAGC IR codes overlay the DAP controller IR codes. JTAGC uses twelve instructions and the DAP uses the remaining four instructions. The outputs of the TAPs (TDO) are multiplexed based on the IR code which is selected. This design is fully JTAG compliant and appears to the JTAG chain as a single TAP.

The following table gives the IR codes for the system JTAG controller.

Table 231. JTAG TEST TAP

Code	JTAG IR
4’b0000	IR_EXTEST
4’b0001	SAMPLE/PRELOAD
4’b0010	IR_HIGHZ
4’b0011	IR_CLAMP
4’b0100	IR_IDCODE
4’b0101	Reserved
4’b0110	Reserved
4’b0111	Reserved
4’b1000	ARM-DAP IR_ABORT
4’b1001	Reserved
4’b1010	ARM-DAP IR_DPACC
4’b1011	ARM-DAP IR_APACC
4’b1100	Reserved
4’b1101	Reserved
4’b1110	ARM-DAP IR_DAP_ID
4’b1111	ARM-DAP BYPASS

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

37.3.2 Debug access port

Debug Access Port (DAP) is a standard ARM CoreSight component. The DAP provides multiple master driving ports, all accessible and controlled through a single external interface port to provide system-wide debug. The DAP Instruction Register codes are listed in the following table.

Table 232. DAP IR Codes

Code	DAP IR
4'b1000	ABORT
4'b1010	DPACC
4'b1011	APACC
4'b1110	IDCODE

The DAP offers AHB and APB master interfaces to access system buses. It also exports the internal DAP bus to allow to extend the access ports as per the system requirement. It offers JTAG AP to allow adding auxiliary JTAG TAPs. For more information on DAP or JTAG devices under JTAG AP, refer to ARM Debug Interface v5 Architecture specification on arm.com

The AHB AP's AHB transfers are burst size of 1 only, no out of order transactions, and no multiple outstanding accesses.

The exported DAP bus is used to host AHB AP (integrated as part of CM23 integrations) and the ISP AP (Debug Mailbox). The MDM-AP implement registers which are used for Test mode control on the device. The selection of different access ports in the DAP is done based on the APSEL value set in the SELECT register of SWJ-DP.

37.4 Debug ROM tables

The debug ROM tables reside in the CM23 core hold information about different debug components and help identify them. In this chip, this table contains entries for CM23 debug components.

Table 233. Debug ROM table

Processor		Start address
CPU ROM	CM23 processor ROM	0xE00FF000

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Chapter 38
Debug Mailbox (DBGMB)

38.1 Overview

DBGMB is a small register-based mailbox accessible by both the AHB and the debug port of the chip.

The purpose of DBGMB is to allow the debugger to communicate with the onboard ROM code through the AHB bus, even when the main CPU AP is disabled.

Using this interface, ROM implements a command-response protocol.

NOTE

In this chapter, the term "DM" and "DBGMB" are interchangeable.

38.1.1 Block diagram



Figure 171. Debug Mailbox -Access Port (DBGMB-AP) block diagram

38.1.2 Features

- Register-based mailbox accessible by both the AHB and the debug port of the chip
- Support password based debug authentication

38.2 Functional description

38.2.1 Debugger mailbox access port (DM-AP)

The DM-AP provides a register-based mailbox accessible by a chip's core through AHB and the device debug port (DP). This port is always enabled. An external debugger communicating through the SWD interface can exchange messages and data with the boot code executing from the ROM.

The boot ROM implements a debug mailbox protocol to interact with tools via the SWD interface.

The debug mailbox protocol has the following features:

- Request-and-response based, where the requests and responses use the same basic structure
- Support for relatively large command and response data
- 32-bit word alignment for all commands and responses
- Support for data above 32 bits via an ACK_TOKEN that moderates the transfer in 32-bit value chunks

38.2.2 Password authentication

Password authentication mechanism is basically a 128-bit password matching: a debug probe writes a 128-bit password into the mailbox. If it matches the preset password, then the mailbox asserts an output “auth_success” signal.

38.3 External signals

Table 234 shows the signals related to the debug process. A trace using the serial wire output has limited bandwidth.

© 2026 NXP B.V. All rights reserved.

Table 234. DBGMB signals

Signal	I/O	Description
SWCLK	I	Serial Wire Debug (SWD) clock. Provides the clock for the SWD logic in SWD mode. SWCLK is the default signal for its pin. At the release of reset, the pin is pulled down internally.
SWDIO	I/O	SWD data input and output. Used by an external debug tool to communicate with and control the part. SWDIO is the default signal of its pin. At the release of reset, the pin is pulled up internally.

38.4 Memory map and register definition

38.4.1 DBGMB register descriptions

38.4.1.1 DBGMB memory map

DebugMailbox0 base address: 4010_1000h

Offset	Register	Width (In bits)	Access	Reset value
0h	Command and Status Word (CSW)	32	RW	0000_0000h
4h	Request Value (REQUEST)	32	RW	0000_0000h
8h	Return Value (RETURN)	32	RW	0000_0000h
80h - 8Ch	Password n (PW0 - PW3)	32	W	See section
FCh	Identification (ID)	32	R	002A_0000h

38.4.1.2 Command and Status Word (CSW)

Offset

Register	Offset
CSW	0h

Function

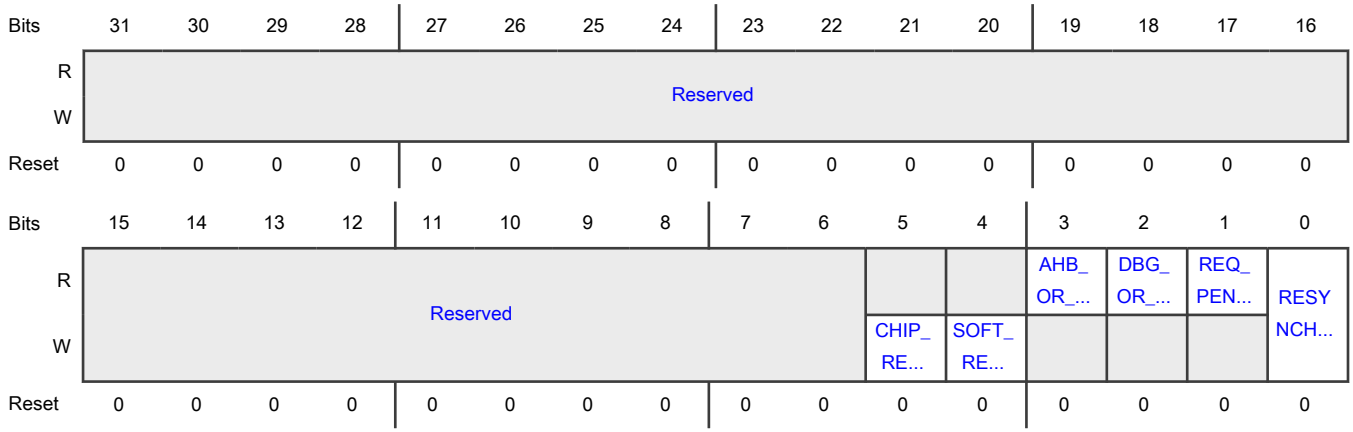
Contains command and status bits to facilitate communication between the debugger and the core.

Both the debugger and the core can write and read this register.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram



Fields

Field	Function
31-6 —	Reserved
5 CHIP_RESET_REQ	Chip Reset Request Causes the chip (but not the DM-AP) to be reset by generating SYSRESET_REQ. This field is write only. 0b - No effect 1b - Reset
4 SOFT_RESET	Soft Reset Resets the DM-AP. This field is write only by the chip. 0b - No effect 1b - Reset
3 AHB_OR_ERR	AHB Overrun Error Indicates whether an AHB overrun has occurred: the chip has overwritten a RETURN value before DBGMB reads the value of Return Value (RETURN). 0b - No overrun 1b - Overrun occurred
2 DBG_OR_ERR	DBGMB Overrun Error Indicates whether a DBGMB overrun has occurred: DBGMB has overwritten a REQUEST value before the chip reads the value of Request Value (REQUEST). 0b - No overrun 1b - Overrun occurred
1	Request Pending

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
REQ_PENDING	Indicates a pending request from DBGMB: a value is waiting to be read by the chip from Request Value (REQUEST). 0b - No request pending 1b - The debugger has written a value into the REQUEST register and it's pending the chip reading REQUEST
0 RESYNCH_REQ	Resynchronization Request Requests a resynchronization. When the debugger writes 1 to this field, AP interface is required to be in reset state, and the core is expected to clear it by writing 1 to [SOFT_RESET]. If this field is not cleared as 0, when the debugger reads Command and Status Word (CSW), the read is always stalled. 0b - No request 1b - Request for resynchronization

38.4.1.3 Request Value (REQUEST)

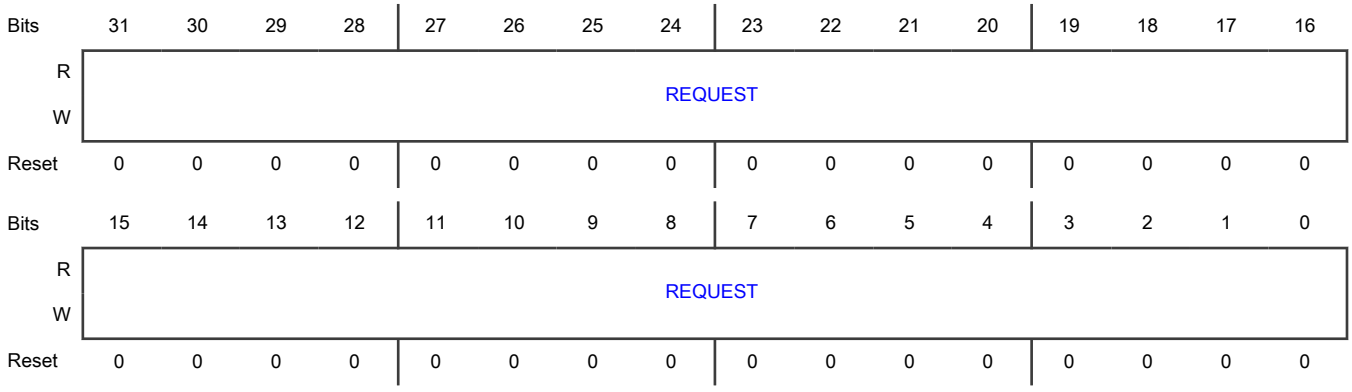
Offset

Register	Offset
REQUEST	4h

Function

Used by DBGMB to send action requests to the chip.

Diagram



All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Fields

Field	Function
31-0 REQUEST	Request Value Indicates the request value. This field reads 0 when no new request is present. External debugger can write and read REQUEST, but the core can only read Request Value (REQUEST). If the debugger writes Request Value (REQUEST) again before the core reads it, CSW[DBG_OR_ERR] is set.

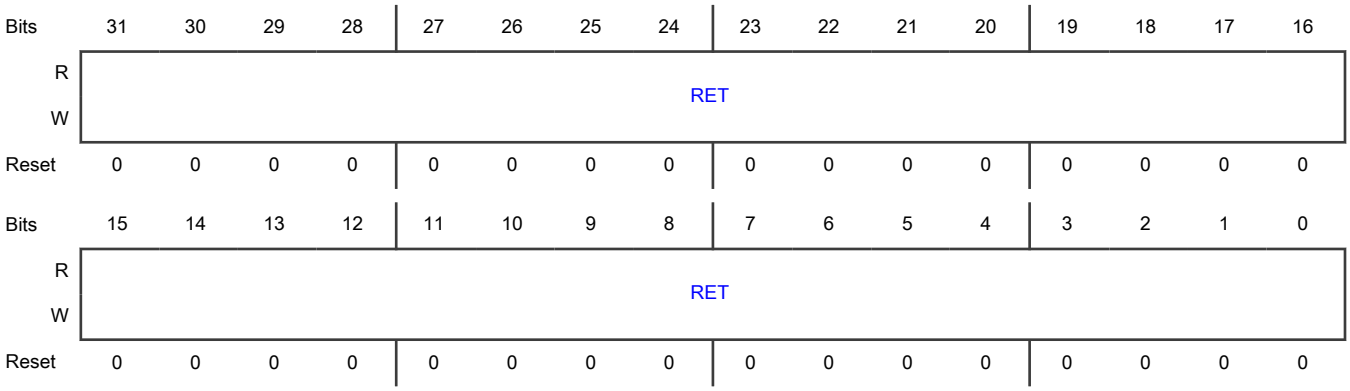
38.4.1.4 Return Value (RETURN)

Offset

Register	Offset
RETURN	8h

Function
Provides the responses from the chip to DBGMB.

Diagram



Fields

Field	Function
31-0 RET	Return Value Indicates the return value, which is a response from the chip to DBGMB. If no new data is present, DBGMB reading is stalled until new data is available. External debugger can only read RETURN, the core can write and read Return Value (RETURN). • If the core writes Return Value (RETURN) again before the debugger reads it, the register value will be updated but CSW[AHB_OR_ERR] is set. • If the debugger reads Return Value (RETURN) before the core writes it, the reading is stalled.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

38.4.1.5 Password n (PW0 - PW3)

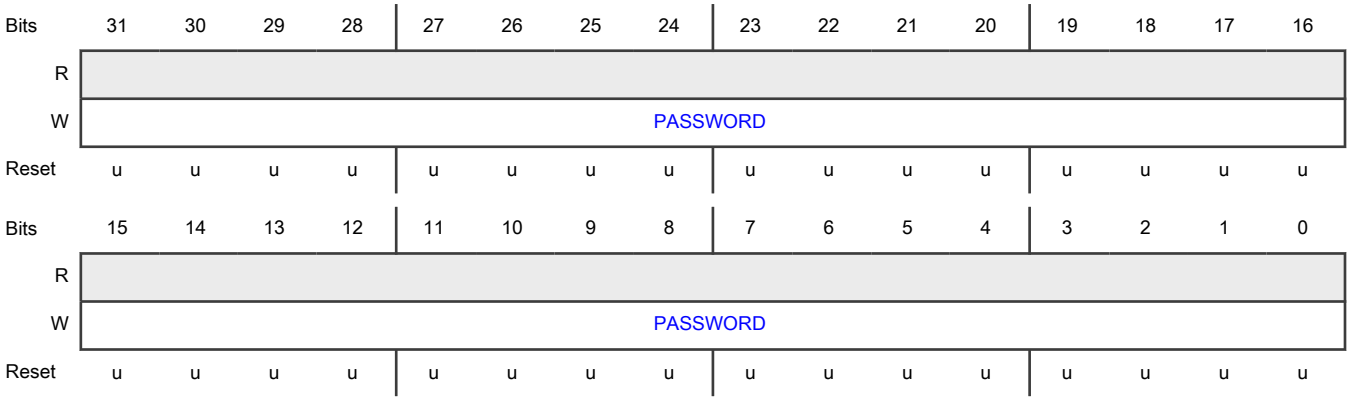
Offset

Register	Offset
PW0	80h
PW1	84h
PW2	88h
PW3	8Ch

Function

Contains a 32-bit segment of the 128-bit password. See Password authentication.

Diagram



Fields

Field	Function
31-0 PASSWORD	Password Contains a 32-bit segment of the 128-bit password.

38.4.1.6 Identification (ID)

Offset

Register	Offset
ID	FCh

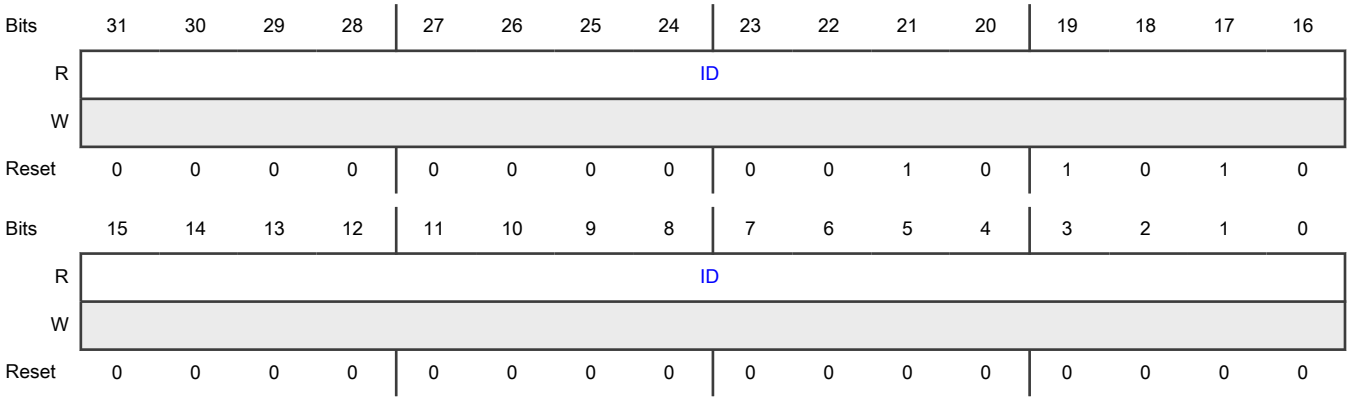
Function

Provides an identification of the DM-AP interface.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram



Fields

Field	Function
31-0 ID	Identification Value Provides an identification of the DM-AP interface. The Arm Debug Interface Specification version 5.0 (ADIv5) requires every AP to implement an AP identification register, at offset FCh. This register is the last register in the AP register space. This ID register is only readable by external tools (a debug dongle) when identifying the Access Port (AP). For the debug mailbox AP, the identification value is 002A_0000h. This register is not readable from on-chip software. If you attempt a read, you receive a value of 0000_0000h.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Chapter 39
Cyclic Redundancy Check (CRC)

39.1 Chip-specific CRC information

Table 235. Reference links to related information

Topic	Related module	Reference
Full description	CRC	CRC
System memory map		Memory map
Clocking		Clock distribution
Power Management		Power Management
Signal multiplexing	Port control	Signal multiplexing

39.1.1 Module instances

This chip has one instance of the CRC module, CRC0.

39.2 Overview

CRC generates 16-bit or 32-bit CRC codes output for error detection. You can calculate these codes for up to 32 bits input data at a time.

CRC provides a programmable polynomial and other parameters to meet 16-bit or 32-bit CRC standards.

39.2.1 Block diagram

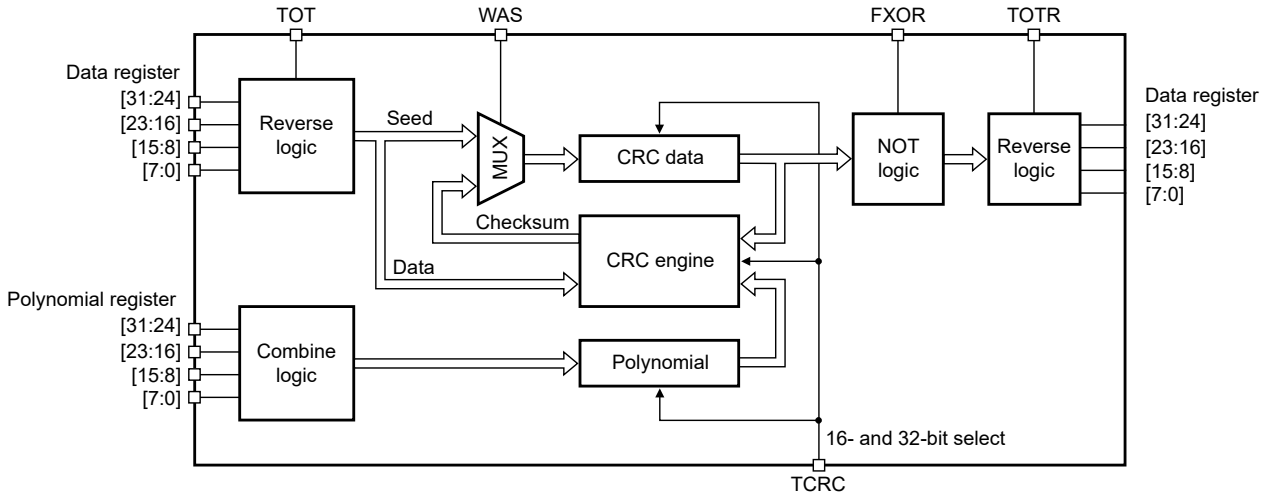


Figure 172. Block diagram

39.2.2 Features

CRC has the following features:

- Hardware CRC generator circuit using 16-bit or 32-bit programmable shift registers

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

- Programmable initial seed value and polynomial
- Transpose of input or output data (CRC result) in bitwise or bytewise (this option is required for certain CRC standards. You cannot perform a bytewise transpose operation when accessing DATA via 8-bit access.)
- Invert final CRC result
- 32-bit CPU register programming interface

39.3 Functional description

39.3.1 Modes of operation

The following sections describe various modes of operation that affect the functionality of CRC: Run mode and Low power mode.

39.3.1.1 Run mode

Run mode is the basic mode of operation.

39.3.1.2 Low-power modes

When the chip enters the low-power mode, the CRC module clock (ipg_clk and ipg_clk_s) is disabled and the in-progress CRC calculation stops. The calculation resumes after the CRC module clock is enabled or the chip exits low power mode via system reset.

NOTE

See the chip-specific power management chapter to know the specific low-power modes supported on your device.

39.3.2 CRC calculations

In 16-bit and 32-bit CRC modes, you can program data values as 8 bits, 16 bits, or 32 bits at a time, provided all bytes are contiguous. Noncontiguous bytes lead to an incorrect CRC calculation.

39.3.2.1 Calculating a 16-bit CRC

Perform these steps to calculate a 16-bit CRC:

1. Write 0 to CTRL[TCRC] to enable 16-bit CRC mode.
2. Program the transpose and complement options in Control (CTRL) as required for the CRC calculation.
3. Write a 16-bit polynomial to GPOLY[LOW].

   GPOLY[HIGH] is not usable in 16-bit CRC mode.
4. Write 1 to CTRL[WAS] to program the seed value.
5. Write a 16-bit seed to DATA[LU] and DATA[LL].

   DATA[HU] and DATA[HL] are not used.
6. Write 0 to CTRL[WAS] to start writing data values.
7. Write data values into DATA[LU], and DATA[LL]

   CRC is calculated on every data value write and the intermediate CRC result is stored back into DATA[LU] and DATA[LL]
8. After writing all the data values, read the final CRC result from DATA[LU] and DATA[LL].

CRC is calculated bytewise and two clocks are required to complete one CRC calculation.

The transpose and complement operations are performed on-the-fly when reading or writing values. See Transpose feature and Result complement for details.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

39.3.2.2 Calculating a 32-bit CRC

Perform these steps to calculate a 32-bit CRC:

1. Write 1 to CTRL[TCRC] to enable 32-bit CRC mode.
2. Program the transpose and complement options in Control (CTRL) as required for CRC calculation. See Transpose feature and Result complement for details.
3. Write a 32-bit polynomial to GPOLY[HIGH] and GPOLY[LOW] .
4. Write 1 to CTRL[WAS] to program the seed value.
5. Write a 32-bit seed to DATA[HU], DATA[HL], DATA[LU], and DATA[LL].
6. Write 0 to CTRL[WAS] to start writing data values.
7. Write data values into DATA[HU], DATA[HL], DATA[LU], and DATA[LL]

   CRC is computed on every data value write and the intermediate CRC result is stored back into DATA .
8. After writing all the values, read the final CRC result from DATA[HU], DATA[HL], DATA[LU], and DATA[LL].

   CRC is calculated bytewise and two clocks are required to complete one CRC calculation.

The transpose and complement operations are performed on-the-fly when reading or writing values. See Transpose feature and Result complement for details.

39.3.3 Transpose feature

Transpose is not enabled by default. However, CRC requires input data and/or final checksum to be transposed. You have an option to configure each transpose operation separately to meet CRC standards. The data is transposed on-the-fly while being read or written.

Some protocols use the little-endian format for data stream to calculate CRC. In this case, transpose flips bits.

39.3.3.1 Types of transpose

CRC provides several types of transpose to flip bits and/or bytes for both writing input data and reading result separately using the CTRL[TOT] and CTRL[TOTR] according to the CRC calculation being used.

The following types of transpose are available for writing to and reading from DATA.

1. CTRL[TOT] or CTRL[TOTR] is 0.

   No transposition occurs.
2. CTRL[TOT] or CTRL[TOTR] is 1.

   Bits in a byte are transposed when bytes are not transposed.

   reg[31:0] becomes {reg[24:31], reg[16:23], reg[8:15], reg[0:7]}.

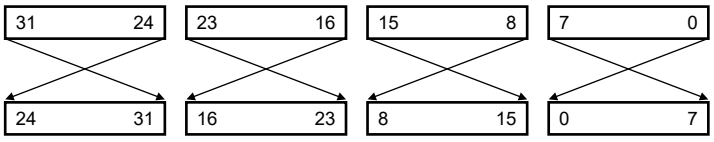


Figure 173. Transpose type 1b

3. CTRL[TOT] or CTRL[TOTR] is 10b.

   Both bits in bytes and bytes are transposed.

   reg[31:0] becomes {reg[0:7], reg[8:15], reg[16:23], reg[24:31]}.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

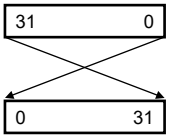


Figure 174. Transpose type 10b

4. CTRL[TOT] or CTRL[TOTR] is 11b.

   Bytes are transposed but bits are not transposed.

   reg[31:0] becomes {reg[7:0], reg[15:8], reg[23:16], reg[31:24]}.

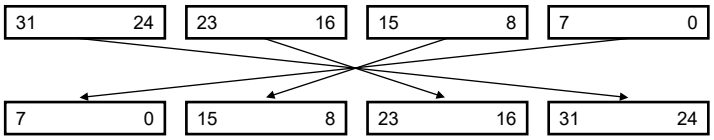


Figure 175. Transpose type 11b

NOTE

For 8-bit and 16-bit write accesses to Data (DATA), the data is transposed with 0s on the unused byte or bytes (taking 32 bits as a whole), but CRC is calculated on the valid byte(s) only. When reading the Data (DATA) for a 16-bit CRC result and using transpose options 10 and 11, the resulting value after transposition resides in DATA[HU] and DATA[HL]. You must account for this situation when reading the 16-bit CRC result, so reading 32 bits is preferred.Additionally, when writing the initial SEED value, the CRC module processes the SEED using the same transpose type configured in CTRL[TOT] as used for normal input data. Therefore, it is recommended to keep CTRL[TOT] configuration consistent for both WAS = 1 (SEED write) and WAS = 0 (normal DATA write) to ensure the SEED value is interpreted correctly.

39.3.4 Result complement

When CTRL[FXOR] = 1, the checksum is complemented. The CRC result complement function outputs the complement of the checksum value stored in Data (DATA) every time Data (DATA) is read. When CTRL[FXOR] = 0, reading Data (DATA) accesses the raw checksum value.

39.3.5 Clocking

Table 236. CRC clocks

Type of clock	Description
Bus clock (ipg_clk / ipg_clk_s)	ipg_clk_s controls the access to the CRC registers. ipg_clk and ipg_clk_s function the CRC module.

39.3.6 Interrupts

This module has no interrupts.

39.4 External signals

There is no CRC signal that connects off chip.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

39.5 Initialization

To enable CRC calculation, you must program:

- CTRL[WAS] .
- Polynomial (GPOLY).
- Parameters for transposition and CRC result inversion in the applicable registers.

Writing 1 to CTRL[WAS] enables you to program the seed value into CRC Data registers.

After writing all the data, you must wait for at least two clock cycles to read the data from CRC Data (DATA) register.

After a CRC calculation completes, you can reinitialize the module for a new CRC computation by again writing 1 to CTRL[WAS] and programming a new, or previously used, seed value. You must set all other parameters before programming the seed value and subsequent data values.

39.6 Use cases

The following tables use the little-endian format.

39.6.1 CTRL programming

The following table shows Control (CTRL) programming for 16-bit CRC.

Table 237. CTRL programming for 16-bit CRC

Algorithm	Polynomial	Seed	Ref in	Ref out	XOR out	CTRL[TOT]	CTRL[TOTR]	CTRL[FXOR]
CRC-16_CCITT_FALSE	1021h	FFFFh	0	0	0000h	0h	0h	0h
CRC-16_ARC	8005h	0000h	1	1	0000h	1h	2h	0h
CRC-16_AUG_CCITT	1021h	1D0Fh	0	0	0000h	0h	0h	0h
CRC-16_BUYPASS	8005h	0000h	0	0	0000h	0h	0h	0h
CRC-16_CCITT_ZERO	1021h	0000h	0	0	0000h	0h	0h	0h
CRC-16_CDMA2000	C867h	FFFFh	0	0	0000h	0h	0h	0h
CRC-16_DDS_110	8005h	800Dh	0	0	0000h	0h	0h	0h
CRC-16_DECT_X	589h	0000h	0	0	0000h	0h	0h	0h
CRC-16_DNP	3D65h	0000h	1	1	FFFFh	1h	2h	1h
CRC-16_EN_13757	3D65h	0000h	0	0	FFFFh	0h	0h	1h
CRC-16_GENIBUS	1021h	FFFFh	0	0	FFFFh	0h	0h	1h
CRC-16_MAXIM	8005h	0000h	1	1	FFFFh	1h	2h	1h
CRC-16_MCRF4XX	1021h	FFFFh	1	1	0000h	1h	2h	0h
CRC-16_RIELLO	1021h	B2AAh	1	1	0000h	1h	2h	0h
CRC-16_T10_DIF	8BB7h	0000h	0	0	0000h	0h	0h	0h
CRC-16_TELEDISK	A097h	0000h	0	0	0000h	0h	0h	0h
CRC-16_TMS37157	1021h	89ECh	1	1	0000h	1h	2h	0h
CRC-16_A	1021h	C6C6h	1	1	0000h	1h	2h	0h

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table 237. CTRL programming for 16-bit CRC (continued)

Algorithm	Polynomial	Seed	Ref in	Ref out	XOR out	CTRL[TOT]	CTRL[TOTR]	CTRL[FXOR]
CRC-16_KERMIT	1021h	0000h	1	1	0000h	1h	2h	0h
CRC-16_MODBUS	8005h	FFFFh	1	1	0000h	1h	2h	0h
CRC-16_X_25	1021h	FFFFh	1	1	FFFFh	1h	2h	1h
CRC-16_XMODEM	1021h	0000h	0	0	0000h	0h	0h	0h

The following table shows Control (CTRL) programming for 32-bit CRC.

Table 238. CTRL programming for 32-bit CRC

Algorithm	Polynomial	Seed	Ref in	Ref out	XOR out	CTRL[TOT]	CTRL[TOTR]	CTRL[FXOR]
CRC-32	04C11DB7h	FFFFFFFFh	1	1	FFFF_FFFFh	1h	2h	1h
CRC-32_BZIP2	04C11DB7h	FFFFFFFFh	0	0	FFFF_FFFFh	0h	0h	1h
CRC-32C	1EDC6F41h	FFFFFFFFh	1	1	FFFF_FFFFh	1h	2h	1h
CRC-32D	A833982Bh	FFFFFFFFh	1	1	FFFF_FFFFh	1h	2h	1h
CRC-32_MPEG-2	04C11DB7h	FFFFFFFFh	0	0	0000_0000h	0h	0h	0h
CRC-32_POSIX	04C11DB7h	00000000h	0	0	FFFF_FFFFh	0h	0h	1h
CRC-32Q	814141ABh	00000000h	0	0	0000_0000h	0h	0h	0h
CRC-32_JAMCRC	04C11DB7h	FFFFFFFFh	1	1	0000_0000h	1h	2h	0h
CRC-32_XFER	000000AFh	00000000h	0	0	0000_0000h	0h	0h	0h

39.6.2 Expected read data fields

The following table shows the expected read data fields for 16-bit CRC.

Table 239. Expected read data fields for 16-bit CRC

Algorithm	Data (DATA)
CRC16_CCITT_FALSE	[31:16] = Unknown[15:0] = Valid data
CRC16_ARC	[31:16] = Valid data [15:0] = Unknown
CRC16_AUG_CCITT	[31:16] = Unknown [15:0] = Valid data
CRC16_BUYPASS	[31:16] = Unknown [15:0] = Valid data
CRC16_CCITT_ZERO	[31:16] = Unknown [15:0] = Valid data
CRC16_CDMA2000	[31:16] = Unknown [15:0] = Valid data
CRC16_DDS_110	[31:16] = Unknown [15:0] = Valid data
CRC16_DECT_X	[31:16] = Unknown [15:0] = Valid data
CRC16_DNP	[31:16] = Valid data [15:0] = Unknown
CRC-16_EN_13757	[31:16] = Unknown [15:0] = Valid data

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table 239. Expected read data fields for 16-bit CRC (continued)

Algorithm	Data (DATA)
CRC-16_GENIBUS	[31:16] = Unknown [15:0] = Valid data
CRC-16_MAXIM	[31:16] = Valid data [15:0] = Unknown
CRC-16_MCRF4XX	[31:16] = Valid data [15:0] = Unknown
CRC-16_RIELLO	[31:16] = Valid data [15:0] = Unknown
CRC-16_T10_DIF	[31:16] = Unknown [15:0] = Valid data
CRC-16_TELEDISK	[31:16] = Unknown [15:0] = Valid data
CRC-16_TMS37157	[31:16] = Valid data [15:0] = Unknown
CRC-16_USB	[31:16] = Valid data [15:0] = Unknown
CRC-16_A	[31:16] = Valid data [15:0] = Unknown
CRC-16_KERMIT	[31:16] = Valid data [15:0] = Unknown
CRC-16_MODBUS	[31:16] = Valid data [15:0] = Unknown
CRC-16_X_25	[31:16] = Valid data [15:0] = Unknown
CRC-16_XMODEM	[31:16] = Unknown [15:0] = Valid data

The following table shows the expected read data fields for 32-bit CRC.

Table 240. Expected read data fields for 32-bit CRC

Algorithm	Data (DATA)
CRC-32	[31:0] = Valid data
CRC-32_BZIP2	[31:0] = Valid data
CRC-32C	[31:0] = Valid data
CRC-32D	[31:0] = Valid data
CRC-32_MPEG-2	[31:0] = Valid data
CRC-32_POSIX	[31:0] = Valid data
CRC-32Q	[31:0] = Valid data
CRC-32_JAMCRC	[31:0] = Valid data
CRC-32_XFER	[31:0] = Valid data

39.7 Memory map and register descriptions

NOTE

You must reconfigure CRC engine in case an IPS transfer error occurs (ips_xfr_err).

The CRC module generates a transfer error in the following cases:

- Write accesses to the register addresses that are not mapped to the peripherals but included in the address spaces of the peripherals.

39.7.1 CRC register descriptions

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

39.7.1.1 CRC memory map

CRC0 base address: 4008_A000h

Offset	Register	Width (In bits)	Access	Reset value
0h	Data (DATA)	32	RW	FFFF_FFFFh
4h	Polynomial (GPOLY)	32	RW	0000_1021h
8h	Control (CTRL)	32	RW	0000_0000h

39.7.1.2 Data (DATA)

Offset

Register	Offset
DATA	0h

Function

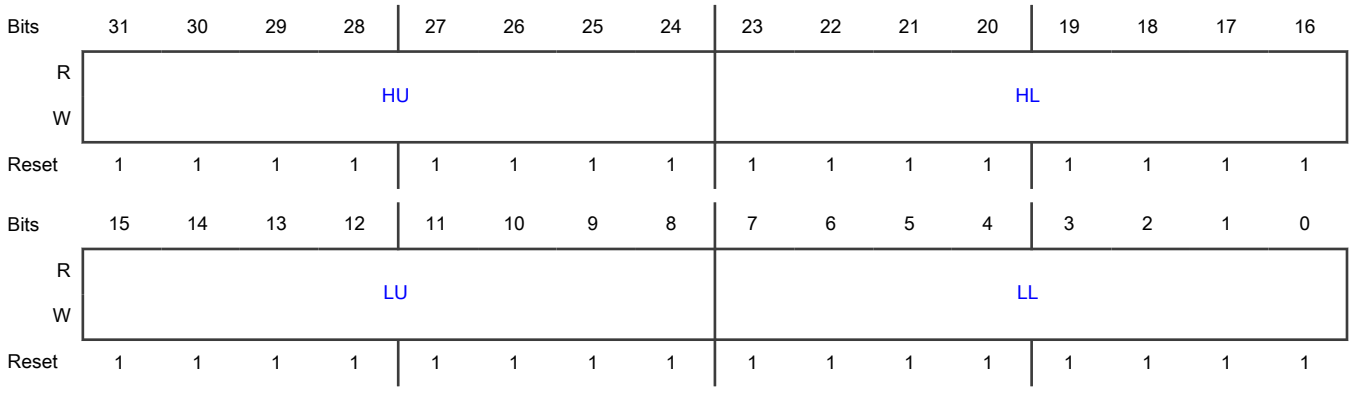
Configures the value of seed, data, and checksum. When CTRL[WAS] = 1, any write to this register is regarded as the seed value. When CTRL[WAS] becomes 0, any write to this register is regarded as data for general CRC calculation.

In 16-bit CRC mode, DATA[HU] and DATA[HL] are not used for programming the seed value, and reads of these fields return an indeterminate value. In 32-bit CRC mode, all fields are used for programming the seed value.

When programming data values, you can program to write 8 bits, 16 bits, or 32 bits in big endian order, provided all bytes are contiguous, with the MSB of data value written first.

After writing all data values, you can read the CRC result from DATA register. In 16-bit CRC mode, the CRC result is available in DATA[LU] and DATA[LL]. In 32-bit CRC mode, all fields contain the result. After writing all data, you must wait for at least two clock cycles to read the data from CRC data (DATA) register.

Diagram



All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Fields

Field	Function
31-24 HU	Upper Part of High Byte Generates CRC checksum in both 16-bit and 32-bit CRC modes if CTRL[WAS] = 0. • In 16-bit CRC mode (CTRL[TCRC] = 0), this field is not used for programming a seed value. • In 32-bit CRC mode (CTRL[TCRC] = 1), the values written to this field are part of the seed value when CTRL[WAS] = 1.
23-16 HL	Lower Part of High Byte Generates CRC checksum in both 16-bit and 32-bit CRC modes if CTRL[WAS] = 0. • In 16-bit CRC mode (CTRL[TCRC] = 0), this field is not used for programming a seed value. • In 32-bit CRC mode (CTRL[TCRC] = 1), the values written to this field are part of the seed value when CTRL[WAS] = 1.
15-8 LU	Upper Part of Low Byte Generates CRC checksum when CTRL[WAS] = 0. When CTRL[WAS] = 1,the values written to this field are part of the seed value.
7-0 LL	Lower Part of Low Byte Generates CRC checksum when CTRL[WAS] = 0. When CTRL[WAS] = 1, the values written to this field are part of the seed value.

39.7.1.3 Polynomial (GPOLY)

Offset

Register	Offset
GPOLY	4h

Function
Configures polynomial value for CRC calculation.

- Sets the upper 16 bits of polynomial that are used only in 32-bit CRC mode. Writes to this field are ignored in 16-bit CRC mode.
- Sets the lower 16 bits of polynomial that are used in both 16-bit and 32-bit CRC modes.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	HIGH															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	LOW															
W																
Reset	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	1

Fields

Field	Function
31-16 HIGH	High Half-Word Writable and readable in 32-bit CRC mode (CTRL[TCRC] = 1). You cannot write to this field in 16-bit CRC mode (CTRL[TCRC] = 0).
15-0 LOW	Low Half-Word Writable and readable in both 16-bit and 32-bit CRC modes.

39.7.1.4 Control (CTRL)

Offset

Register	Offset
CTRL	8h

Function

Sets control for CRC. You must write 1 to the appropriate fields of this register before starting a new CRC calculation, which you can initialize by writing 1 to CTRL[WAS] and then writing the seed into DATA.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	TOT		TOTR		0	FXOR	WAS	TCRC	0							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Fields

Field	Function
31-30 TOT	Transpose Type for Write Sets transpose type for the values written to DATA. See Transpose feature for the available transpose options. 00b - No transposition 01b - Bits in bytes are transposed, but bytes are not transposed. 10b - Both bits in bytes and bytes are transposed. 11b - Only bytes are transposed, no bits in a byte are transposed.
29-28 TOTR	Transpose Type for Read Sets transpose type for the values read from DATA . See Transpose feature for the available transpose options. 00b - No transposition 01b - Bits in bytes are transposed, but bytes are not transposed. 10b - Both bits in bytes and bytes are transposed. 11b - Only bytes are transposed, no bits in a byte are transposed.
27 —	Reserved
26 FXOR	Complement Read of CRC Data Register Enables on-the-fly complementing of read data. Some CRC protocols require the final checksum to be XORed with FFFFFFFFh or FFFFh. 0b - Disables XOR on reading data. 1b - Inverts or complements the read value of the CRC Data.
25 WAS	Write as Seed Specifies whether writes to DATA are data values or seed values. When this field = 1, the value that you write to is considered as seed value. When this field = 0, the value that you write to is considered as data for CRC calculation. 0b - Data values 1b - Seed values
24 TCRC	TCRC Defines the width of CRC. 0b - 16 bits 1b - 32 bits
23-0 —	Reserved

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Chapter 40
Memory Block Checker (MBC)

40.1 Chip-specific MBC information

Table 241. Reference links to related information

Topic	Related module	Reference
Full description	MBC (TRDC)	MBC
System memory map		Memory map
Clocking		Clock distribution
Power Management		Power Management
Signal multiplexing	Port control	Signal multiplexing

The Memory Block Checker (MBC) is a sub-module of Trusted Resource Domain Controller (TRDC). TRDC is typically distributed across various sub-modules like MGR, DAC, MBC and MRC. However, sub-modules MGR, DAC, and MRC do not exist on this device and any references to them in TRDC chapter should be ignored. This chapter covers TRDC sub-module MBC and its features in detail.

40.1.1 Module instances

There is one MBC module and one Domain on this chip. m and d in MBCm_DOMd_BLK_CFG_W are 0. The MBC is instanced between Flash Memory Controller (FMC) and AHB Matrix. MBC can control the Read, Write and Execute permission of each flash block, and the configuration can be locked.

40.1.2 MBC memory blocks

The granularity of MBC is described in the following table.

MBC0_MEMn	Memory	Block size
MEM0	Main flash	8 KB
MEM1	IFR0	2 KB
MEM2	IFR1	2 KB

NOTE

MBC0_MEM3_GLBCFG[CLRE] is used to clear violation interrupt of MEM0/1/2.

40.1.3 Security considerations

MBC on this chip can't distinguish secure privilege, secure user, non-secure privilege, and non- secure user, when there is an access to flash, and MBC treat all access to flash as secure user level. Because of it, SUR/SUW/SUX bits in MBC Global Access Control registers control the permission to flash. User should not set the NonSecure Enable (NSE) bits in MBC Global Access Control registers.

MBC controls the read, write and execute permission to flash. If an access doesn't have the permission to the flash region, MBC will generate interrupt. The domainX error registers are physically in the block checker.

NOTE

TZ-M feature of TRDC is not supported on this device.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

40.2 Overview

The Memory Block Checker (MBC) implements access controls for on-chip internal memories based on a fixed-sized block format. MBC provides hardware access control for system bus references targeted at on-chip memory spaces.

Using the pre-programmed region descriptors with fixed-sized block format and their associated access rights, the MBC concurrently monitors system bus transactions and evaluates the appropriateness of each transfer. Memory references with sufficient access control rights are completed, while the references that are not mapped to any region descriptor or have insufficient rights are terminated with an access error response.

NOTE

The terminology in this chapter has been updated to align with NXP's inclusive language standards, as shown in the table below.

Table 242. Updated terms

Updated term	Deprecated term
Initiator	Master
Target	Slave

40.2.1 Block diagram

The Memory Block Checker (MBC) implements access controls for on-chip internal memories and peripherals based on a fixed-sized block format.

See Figure 176 for MBC block diagram focusing on topology and connections.

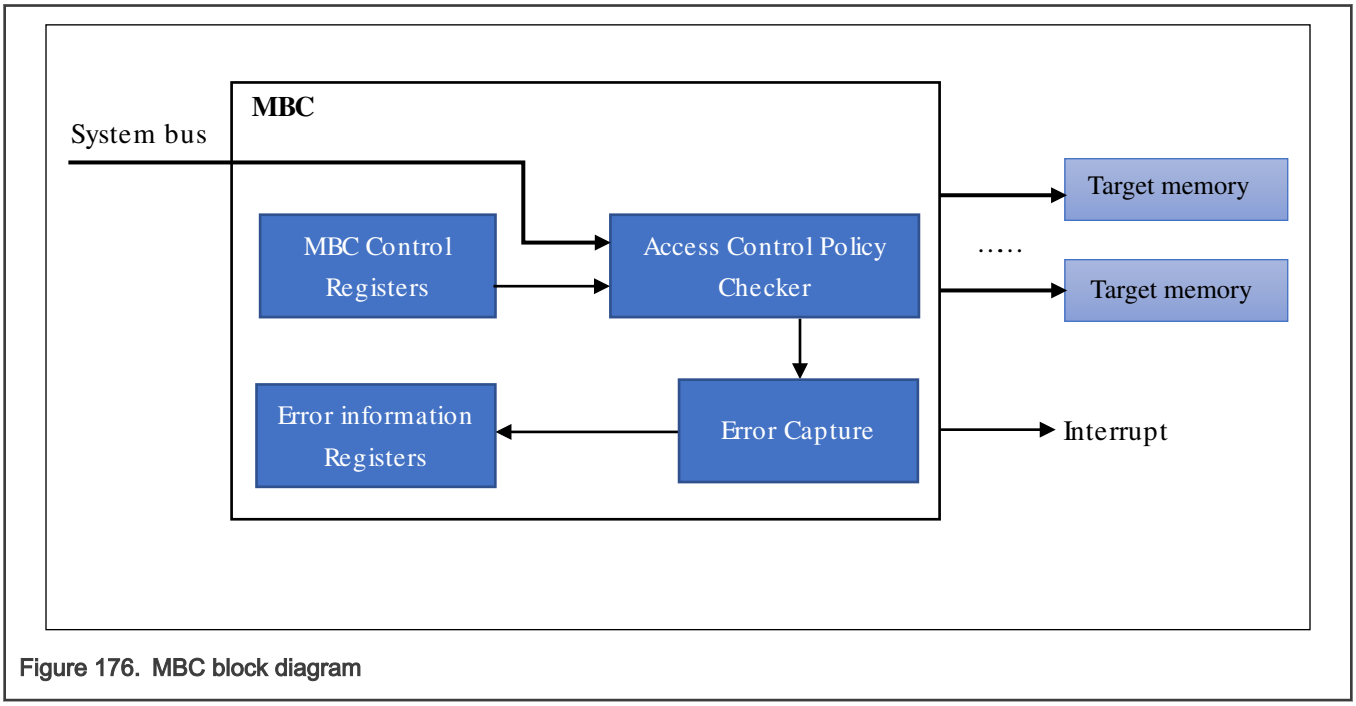


Figure 176. MBC block diagram

40.2.2 Features

Defines access rights to targets defined in MBCs for on-chip memories.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

40.3 Functional description

This section provides more details on the operation and implementation of MBC.

40.3.1 Memory Block Checker (MBC)

The Memory Block Checker provides access control for system bus references targeted to on-chip internal memories . Using programmed block configuration registers which define the access rights per block, the MBC concurrently monitors system bus transactions and evaluates the appropriateness of each transfer. Memory references that have sufficient access rights are allowed to complete, while references that have insufficient rights are terminated with an access error response.

Refer to the chip-specific section to determine the number and size of blocks for each sub memory supported on this device.

Each MBCm_DOMd_BLK_CFG_W contains eight access control structures. Each structure is comprised of a 3-bit MBACSEL (Memory Block Access Control Select). The MBACSEL field selects a MBCm_MEMN_GLBACr register, which controls the read/write/execute/lock access to the corresponding block. Below table provides the eight possible access control combinations. In case you are looking for different access combination, such as write and execute allowed but read blocked, you can modify one of the unlocked configurations registers to 0x0000_3300.

Table below depicts eight possible access control combinations.

Table 243. Access control combinations

GAC Index	Register name	Read	Write	Execute	Lock	Description
0	MBC0_MEMN_GLBAC0 (0x0000_6600)	1	1	0	0	**Data flash and unlocked** (RW__). Default flash behavior with read and write access only. Instruction fetches (execute access) are blocked and generates access violation. Blocks assigned with this access control index can be changed to another index by software. This configuration register is unlocked and updatable by application.
1	MBC0_MEMN_GLBAC1 (0x8000_6600)	1	1	0	1	**Data flash and locked** (RW_L). Read and write operations are allowed but execution is blocked. Blocks assigned with this access control index cannot be changed to another index until next reset.
2	MBC0_MEMN_GLBAC2 (0x8000_5500)	1	0	1	1	**Read only Memory (ROM) and locked** (R_XL) Read & execute operations are allowed but write is blocked. Blocks assigned with this access control index cannot be changed to another index until next reset.
3	MBC0_MEMN_GLBAC3 (0x8000_4400)	1	0	0	1	**Data Read only memory (DROM) and locked** (R__L) Read operations is allowed but write & execute are blocked. Blocks assigned with this access control index cannot be changed to another index until next reset.
4	MBC0_MEMN_GLBAC4 (0x0000_5500)	1	0	1	0	**Read-Only-Memory (ROM) and unlocked** (R_X_)

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table 243. Access control combinations (continued)

GAC Index	Register name	Read	Write	Execute	Lock	Description
						Read & execute operations are allowed but write is blocked. Blocks assigned with this access control index can be changed to another index by software. This configuration register is unlocked and updatable by application.
5	MBC0_MEMN_GLBAC5 (0x0000_1100)	0	0	1	0	**eXecute-only Memory (XOM) and unlocked** (__X_) Execute operations are allowed but read & write are blocked. Blocks assigned with this access control index can be changed to another index by software. This configuration register is unlocked and updatable by application.
6	MBC0_MEMN_GLBAC6 (0x8000_1100)	0	0	1	1	**eXecute-Only-Memory (XOM) and locked** (__XL) Execute operations are allowed but read & write are blocked. Blocks assigned with this access control index cannot be changed to another index until next reset.
7	MBC0_MEMN_GLBAC7 (0x8000_0000)	0	0	0	1	**Hidden** (___L) Read, Write & execute operations are blocked. Once this access control index is assigned to a block, the memory range associated with block is hidden until next reset.

GLBAC1-7 also have a lock bit MBCm_MEMn_GLBACr[LK]. When LK=1, the GLBACr register is read-only until the next reset. Furthermore, if MBCm_DOMd_BLK_CFG_W[MBACSEL] selects a GLBAC that is locked, the 3-bit MBCACSEL field is also locked until the next reset. GLBAC0 cannot be locked.

40.3.1.1 Memory block hit determination

MBCm_MEMs_GLBCFG[NBLK, SIZE_LOG2] is used to determine which bits of the address correspond to block number.

```
block_n_hit = (addr[MSB:LSB] == n)
where LSB=SIZE_LOG2
MSB=f{NBLKS,LSB}
```

Note that the block hit determination is based only on the address comparison of the first byte being accessed; that is, the MBC does not check the size of the access to make sure it entirely fits within the region.

40.3.1.2 Memory block access evaluation

For a block n hit, the MBC logic evaluates the access rights defined by the MBCm_DOMd_BLK_CFG_Ww registers. The block number hit selects the appropriate MBCm_DOMd_BLK_CFG_Ww register to use in the access evaluation. The {R,W,X},

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

nonsecure and privilege attributes of the access are compared with the MBC global access control policy defined in MBCm_DOMd_BLK_CFG_W[MBACSEL, NSE] fields.

MBC access evaluation terminates the access with a bus error and reports an access error for only one condition - when the access doesn't have sufficient access rights. By nature, the block checker can only hit in ONE block, and by implementation, there are no "miss" accesses. Accesses outside of the range covered by the MBC are not sent to the MBC. There also aren't any individual block valid bits.

40.3.2 Interrupts

This module outputs an interrupt signal which can be connected to the system's interrupt controller. Please check chip-specific interrupt assignment for details. Interrupt is asserted on detection of access violation by any checker, and it remains asserted until MBCn_MEM3_GLBCFG[CLRE] is cleared. This interrupt is in addition to interrupt generated by initiator after receiving bus error due to access violation by memory block checker . So, system may receive interrupt through different channels but from the same access violation at checker.

40.4 External signals

This module has no external signals.

40.5 Register descriptions

Unless noted otherwise, the programming model registers can be accessed via 8-, 16- or 32-bit reads and 32-bit write references. Attempted accesses in a different operating mode, using unsupported write data sizes, writes to read-only resources, or access to reserved spaces are terminated with an error unless noted otherwise.

40.5.1 MBC register descriptions

40.5.1.1 MBC memory map

MBC0__MBC base address: 4008_E000h

Offset	Register	Width (In bits)	Access	Reset value
0h	MBC Global Configuration Register (MBC0_MEM0_GLBCFG)	32	R	000D_0010h
4h	MBC Global Configuration Register (MBC0_MEM1_GLBCFG)	32	R	000B_0010h
8h	MBC Global Configuration Register (MBC0_MEM2_GLBCFG)	32	R	000B_0004h
Ch	MBC Global Configuration Register (MBC0_MEM3_GLBCFG)	32	RW	0000_0000h
20h	MBC Global Access Control (MBC0_MEMN_GLBAC0)	32	RW	0000_6600h
24h	MBC Global Access Control (MBC0_MEMN_GLBAC1)	32	RW	8000_6600h
28h	MBC Global Access Control (MBC0_MEMN_GLBAC2)	32	RW	8000_5500h
2Ch	MBC Global Access Control (MBC0_MEMN_GLBAC3)	32	RW	8000_4400h
30h	MBC Global Access Control (MBC0_MEMN_GLBAC4)	32	RW	0000_5500h
34h	MBC Global Access Control (MBC0_MEMN_GLBAC5)	32	RW	0000_1100h
38h	MBC Global Access Control (MBC0_MEMN_GLBAC6)	32	RW	8000_1100h
3Ch	MBC Global Access Control (MBC0_MEMN_GLBAC7)	32	RW	8000_0000h

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Offset	Register	Width (In bits)	Access	Reset value
40h	MBC Memory Block Configuration Word (MBC0_DOM0_MEM0_BLK_CFG_W0)	32	RW	0000_0000h
44h	MBC Memory Block Configuration Word (MBC0_DOM0_MEM0_BLK_CFG_W1)	32	RW	0000_0000h
180h	MBC Memory Block Configuration Word (MBC0_DOM0_MEM1_BLK_CFG_W0)	32	RW	0000_0000h
184h	MBC Memory Block Configuration Word (MBC0_DOM0_MEM1_BLK_CFG_W1)	32	RW	0000_0000h
1A8h	MBC Memory Block Configuration Word (MBC0_DOM0_MEM2_BLK_CFG_W0)	32	RW	0000_0000h

40.5.1.2 MBC Global Configuration Register (MBC0_MEM0_GLBCFG - MBC0_MEM3_GLBCFG)

Offset

Register	Offset
MBC0_MEM0_GLBCFG	0h
MBC0_MEM1_GLBCFG	4h
MBC0_MEM2_GLBCFG	8h
MBC0_MEM3_GLBCFG	Ch

Function

These MBC global configuration read-only registers contain information on the MBC's hardware configuration. Specifically, it defines the number of memory blocks and the size of each block in each MBC mem (r).

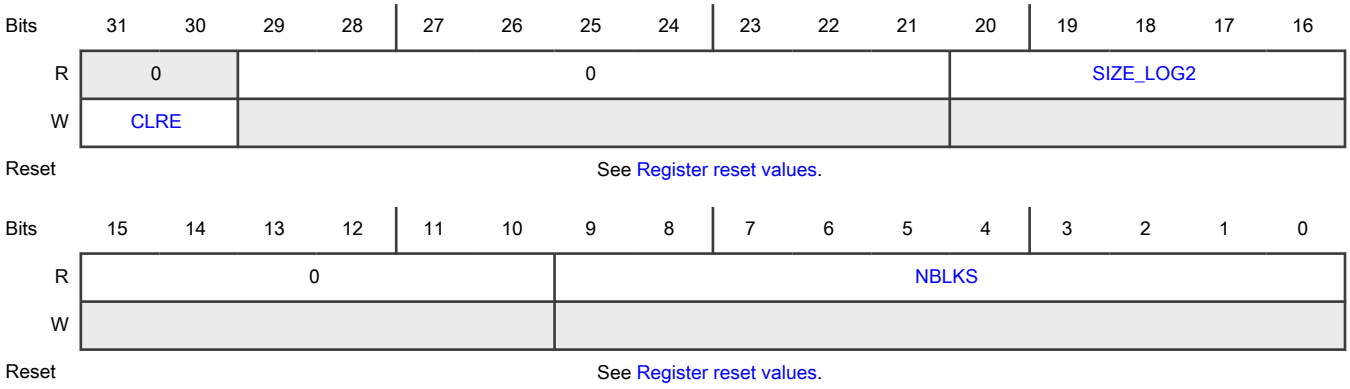
NOTE

An all zero value (SIZE_LOG2 = 0, NBLKS = 0) indicates a non-existent memory. For the number of memories implemented in the MBCs, refer to chip-specific information

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram



Register reset values

Register	Reset value
MBC0_MEM0_GLBCFG	000D_0010h
MBC0_MEM1_GLBCFG	000B_0010h
MBC0_MEM2_GLBCFG	000B_0004h
MBC0_MEM3_GLBCFG	0000_0000h

Fields

Field	Function
31-30 CLRE	Clear Error This 2-bit, write-only field controls the clearing of an access violation error and any interrupt due to error, on writing 01b to this field. A write of any value other than 01b has no effect. **NOTE** This field is not supported in every instance. The following table includes only supported registers.
29-21 —	Reserved
20-16 SIZE_LOG2	Log2 size per block For example SIZE_LOG2=0x0C is 2^12=4 KB blocks.

Instance	Field supported in	Field not supported in
MBC0__MBC	MBC0_MEM3_GLBCFG	MBC0_MEM0_GLBCFG– MBC0_MEM2_GLBCFG

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
15-10 —	Reserved
9-0 NBLKS	Number of blocks in this memory

40.5.1.3 MBC Global Access Control (MBC0_MEMN_GLBAC0)

Offset

Register	Offset
MBC0_MEMN_GLBAC0	20h

Function

These fully programmable R/W fields define the R/W/X (read/write/execute) access control flags with each of the 4 supported operating modes: SecurePriv, SecureUser, NonsecurePriv, Nonsecure User.

NOTE

The lock field (LK) in MBC0_MEMN_GLBAC1, MBC0_MEMN_GLBAC2, MBC0_MEMN_GLBAC3, MBC0_MEMN_GLBAC6 and MBC0_MEMN_GLBAC7 registers are set to lock state (LK= 1) at reset. Hence these registers cannot be changed and blocks that are assigned to one of these global access control indexes cannot change the access policy index until next reset.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	SPR	SPW	SPX	0	SUR	SUW	SUX	0	NPR	NPW	NPX	0	NUR	NUW	NUX
W																
Reset	0	1	1	0	0	1	1	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-15	Reserved

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
—	
14 SPR	SecurePriv Read SecurePriv read access control flag 0b - Read access is not allowed in Secure Privilege mode. 1b - Read access is allowed in Secure Privilege mode.
13 SPW	SecurePriv Write SecurePriv write access control flag 0b - Write access is not allowed in Secure Privilege mode. 1b - Write access is allowed in Secure Privilege mode.
12 SPX	SecurePriv Execute SecurePriv execute access control flag 0b - Execute access is not allowed in Secure Privilege mode. 1b - Execute access is allowed in Secure Privilege mode.
11 —	Reserved
10 SUR	SecureUser Read SecureUser read access control flag 0b - Read access is not allowed in Secure User mode. 1b - Read access is allowed in Secure User mode.
9 SUW	SecureUser Write SecureUser write access control flag 0b - Write access is not allowed in Secure User mode. 1b - Write access is allowed in Secure User mode.
8 SUX	SecureUser Execute SecureUser execute access control flag 0b - Execute access is not allowed in Secure User mode. 1b - Execute access is allowed in Secure User mode.
7 —	Reserved
6	NonsecurePriv Read NonsecurePriv read access control flag

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
NPR	0b - Read access is not allowed in Nonsecure Privilege mode. 1b - Read access is allowed in Nonsecure Privilege mode.
5 NPW	NonsecurePriv Write NonsecurePriv write access control flag 0b - Write access is not allowed in Nonsecure Privilege mode. 1b - Write access is allowed in Nonsecure Privilege mode.
4 NPX	NonsecurePriv Execute NonsecurePriv execute access control flag 0b - Execute access is not allowed in Nonsecure Privilege mode. 1b - Execute access is allowed in Nonsecure Privilege mode.
3 —	Reserved
2 NUR	NonsecureUser Read NonsecureUser read access control flag 0b - Read access is not allowed in Nonsecure User mode. 1b - Read access is allowed in Nonsecure User mode.
1 NUW	NonsecureUser Write NonsecureUser write access control flag 0b - Write access is not allowed in Nonsecure User mode. 1b - Write access is allowed in Nonsecure User mode.
0 NUX	NonsecureUser Execute NonsecureUser execute access control flag 0b - Execute access is not allowed in Nonsecure User mode. 1b - Execute access is allowed in Nonsecure User mode.

40.5.1.4 MBC Global Access Control (MBC0_MEMN_GLBAC1)

Offset

Register	Offset
MBC0_MEMN_GLBAC1	24h

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Function

These fully programmable R/W fields define the R/W/X (read/write/execute) access control flags with each of the 4 supported operating modes: SecurePriv, SecureUser, NonsecurePriv, Nonsecure User. The lock bit provides the ability to lock these control access flags and to lock the MBC_BLK_CFG[MBACSEL] field when a locked MBC_MEMN_GLBAC register is selected.

This register is not writable.

NOTE

The lock field (LK) in MBC0_MEMN_GLBAC1, MBC0_MEMN_GLBAC2, MBC0_MEMN_GLBAC3, MBC0_MEMN_GLBAC6 and MBC0_MEMN_GLBAC7 registers are set to lock state (LK= 1) at reset. Hence these registers cannot be changed and blocks that are assigned to one of these global access control indexes cannot change the access policy index until next reset.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	LK	0														
W																
Reset	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	SPR	SPW	SPX	0	SUR	SUW	SUX	0	NPR	NPW	NPX	0	NUR	NUW	NUX
W																
Reset	0	1	1	0	0	1	1	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31 LK	LOCK This 1-bit field provides a mechanism to limit writes to the this register to protect its contents. Once set, this bit remains asserted until the next reset. 0b - This register is not locked and can be altered. 1b - This register is locked and cannot be altered.
30-15 —	Reserved
14 SPR	SecurePriv Read SecurePriv read access control flag 0b - Read access is not allowed in Secure Privilege mode. 1b - Read access is allowed in Secure Privilege mode.
13 SPW	SecurePriv Write SecurePriv write access control flag

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	0b - Write access is not allowed in Secure Privilege mode. 1b - Write access is allowed in Secure Privilege mode.
12 SPX	SecurePriv Execute SecurePriv execute access control flag 0b - Execute access is not allowed in Secure Privilege mode. 1b - Execute access is allowed in Secure Privilege mode.
11 —	Reserved
10 SUR	SecureUser Read SecureUser read access control flag 0b - Read access is not allowed in Secure User mode. 1b - Read access is allowed in Secure User mode.
9 SUW	SecureUser Write SecureUser write access control flag 0b - Write access is not allowed in Secure User mode. 1b - Write access is allowed in Secure User mode.
8 SUX	SecureUser Execute SecureUser execute access control flag 0b - Execute access is not allowed in Secure User mode. 1b - Execute access is allowed in Secure User mode.
7 —	Reserved
6 NPR	NonsecurePriv Read NonsecurePriv read access control flag 0b - Read access is not allowed in Nonsecure Privilege mode. 1b - Read access is allowed in Nonsecure Privilege mode.
5 NPW	NonsecurePriv Write NonsecurePriv write access control flag 0b - Write access is not allowed in Nonsecure Privilege mode. 1b - Write access is allowed in Nonsecure Privilege mode.
4	NonsecurePriv Execute

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
NPX	NonsecurePriv execute access control flag 0b - Execute access is not allowed in Nonsecure Privilege mode. 1b - Execute access is allowed in Nonsecure Privilege mode.
3 —	Reserved
2 NUR	NonsecureUser Read NonsecureUser read access control flag 0b - Read access is not allowed in Nonsecure User mode. 1b - Read access is allowed in Nonsecure User mode.
1 NUW	NonsecureUser Write NonsecureUser write access control flag 0b - Write access is not allowed in Nonsecure User mode. 1b - Write access is allowed in Nonsecure User mode.
0 NUX	NonsecureUser Execute NonsecureUser execute access control flag 0b - Execute access is not allowed in Nonsecure User mode. 1b - Execute access is allowed in Nonsecure User mode.

40.5.1.5 MBC Global Access Control (MBC0_MEMN_GLBAC2)

Offset

Register	Offset
MBC0_MEMN_GLBAC2	28h

Function

These fully programmable R/W fields define the R/W/X (read/write/execute) access control flags with each of the 4 supported operating modes: SecurePriv, SecureUser, NonsecurePriv, Nonsecure User. The lock bit provides the ability to lock these control access flags and to lock the MBC_BLK_CFG[MBACSEL] field when a locked MBC_MEMN_GLBAC register is selected.

This register is not writable.

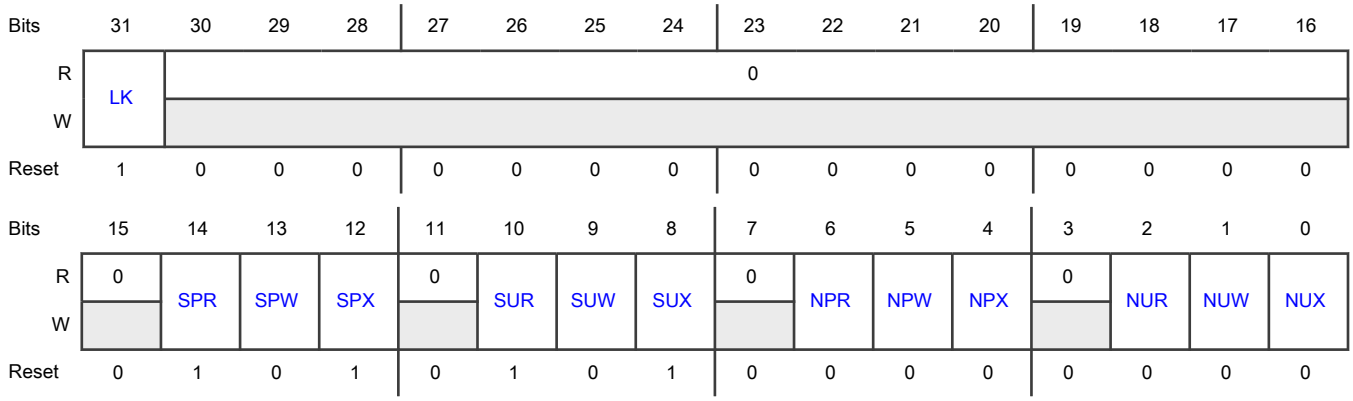
NOTE

The lock field (LK) in MBC0_MEMN_GLBAC1, MBC0_MEMN_GLBAC2, MBC0_MEMN_GLBAC3, MBC0_MEMN_GLBAC6 and MBC0_MEMN_GLBAC7 registers are set to lock state (LK= 1) at reset. Hence these registers cannot be changed and blocks that are assigned to one of these global access control indexes cannot change the access policy index until next reset.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram



Fields

Field	Function
31 LK	LOCK This 1-bit field provides a mechanism to limit writes to the this register to protect its contents. Once set, this bit remains asserted until the next reset. 0b - This register is not locked and can be altered. 1b - This register is locked and cannot be altered.
30-15 —	Reserved
14 SPR	SecurePriv Read SecurePriv read access control flag 0b - Read access is not allowed in Secure Privilege mode. 1b - Read access is allowed in Secure Privilege mode.
13 SPW	SecurePriv Write SecurePriv write access control flag 0b - Write access is not allowed in Secure Privilege mode. 1b - Write access is allowed in Secure Privilege mode.
12 SPX	SecurePriv Execute SecurePriv execute access control flag 0b - Execute access is not allowed in Secure Privilege mode. 1b - Execute access is allowed in Secure Privilege mode.
11 —	Reserved
10	SecureUser Read

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
SUR	SecureUser read access control flag 0b - Read access is not allowed in Secure User mode. 1b - Read access is allowed in Secure User mode.
9 SUW	SecureUser Write SecureUser write access control flag 0b - Write access is not allowed in Secure User mode. 1b - Write access is allowed in Secure User mode.
8 SUX	SecureUser Execute SecureUser execute access control flag 0b - Execute access is not allowed in Secure User mode. 1b - Execute access is allowed in Secure User mode.
7 —	Reserved
6 NPR	NonsecurePriv Read NonsecurePriv read access control flag 0b - Read access is not allowed in Nonsecure Privilege mode. 1b - Read access is allowed in Nonsecure Privilege mode.
5 NPW	NonsecurePriv Write NonsecurePriv write access control flag 0b - Write access is not allowed in Nonsecure Privilege mode. 1b - Write access is allowed in Nonsecure Privilege mode.
4 NPX	NonsecurePriv Execute NonsecurePriv execute access control flag 0b - Execute access is not allowed in Nonsecure Privilege mode. 1b - Execute access is allowed in Nonsecure Privilege mode.
3 —	Reserved
2 NUR	NonsecureUser Read NonsecureUser read access control flag 0b - Read access is not allowed in Nonsecure User mode. 1b - Read access is allowed in Nonsecure User mode.

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
1 NUW	NonsecureUser Write NonsecureUser write access control flag 0b - Write access is not allowed in Nonsecure User mode. 1b - Write access is allowed in Nonsecure User mode.
0 NUX	NonsecureUser Execute NonsecureUser execute access control flag 0b - Execute access is not allowed in Nonsecure User mode. 1b - Execute access is allowed in Nonsecure User mode.

40.5.1.6 MBC Global Access Control (MBC0_MEMN_GLBAC3)

Offset

Register	Offset
MBC0_MEMN_GLBAC3	2Ch

Function

These fully programmable R/W fields define the R/W/X (read/write/execute) access control flags with each of the 4 supported operating modes: SecurePriv, SecureUser, NonsecurePriv, Nonsecure User. The lock bit provides the ability to lock these control access flags and to lock the MBC_BLK_CFG[MBACSEL] field when a locked MBC_MEMN_GLBAC register is selected.

This register is not writable.

NOTE

The lock field (LK) in MBC0_MEMN_GLBAC1, MBC0_MEMN_GLBAC2, MBC0_MEMN_GLBAC3, MBC0_MEMN_GLBAC6 and MBC0_MEMN_GLBAC7 registers are set to lock state (LK= 1) at reset. Hence these registers cannot be changed and blocks that are assigned to one of these global access control indexes cannot change the access policy index until next reset.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	LK	0														
W																
Reset	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	SPR	SPW	SPX	0	SUR	SUW	SUX	0	NPR	NPW	NPX	0	NUR	NUW	NUX
W																
Reset	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Fields

Field	Function
31 LK	LOCK This 1-bit field provides a mechanism to limit writes to the this register to protect its contents. Once set, this bit remains asserted until the next reset. 0b - This register is not locked and can be altered. 1b - This register is locked and cannot be altered.
30-15 —	Reserved
14 SPR	SecurePriv Read SecurePriv read access control flag 0b - Read access is not allowed in Secure Privilege mode. 1b - Read access is allowed in Secure Privilege mode.
13 SPW	SecurePriv Write SecurePriv write access control flag 0b - Write access is not allowed in Secure Privilege mode. 1b - Write access is allowed in Secure Privilege mode.
12 SPX	SecurePriv Execute SecurePriv execute access control flag 0b - Execute access is not allowed in Secure Privilege mode. 1b - Execute access is allowed in Secure Privilege mode.
11 —	Reserved
10 SUR	SecureUser Read SecureUser read access control flag 0b - Read access is not allowed in Secure User mode. 1b - Read access is allowed in Secure User mode.
9 SUW	SecureUser Write SecureUser write access control flag 0b - Write access is not allowed in Secure User mode. 1b - Write access is allowed in Secure User mode.
8 SUX	SecureUser Execute SecureUser execute access control flag

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	0b - Execute access is not allowed in Secure User mode. 1b - Execute access is allowed in Secure User mode.
7 —	Reserved
6 NPR	NonsecurePriv Read NonsecurePriv read access control flag 0b - Read access is not allowed in Nonsecure Privilege mode. 1b - Read access is allowed in Nonsecure Privilege mode.
5 NPW	NonsecurePriv Write NonsecurePriv write access control flag 0b - Write access is not allowed in Nonsecure Privilege mode. 1b - Write access is allowed in Nonsecure Privilege mode.
4 NPX	NonsecurePriv Execute NonsecurePriv execute access control flag 0b - Execute access is not allowed in Nonsecure Privilege mode. 1b - Execute access is allowed in Nonsecure Privilege mode.
3 —	Reserved
2 NUR	NonsecureUser Read NonsecureUser read access control flag 0b - Read access is not allowed in Nonsecure User mode. 1b - Read access is allowed in Nonsecure User mode.
1 NUW	NonsecureUser Write NonsecureUser write access control flag 0b - Write access is not allowed in Nonsecure User mode. 1b - Write access is allowed in Nonsecure User mode.
0 NUX	NonsecureUser Execute NonsecureUser execute access control flag 0b - Execute access is not allowed in Nonsecure User mode. 1b - Execute access is allowed in Nonsecure User mode.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

40.5.1.7 MBC Global Access Control (MBC0_MEMN_GLBAC4)

Offset

Register	Offset
MBC0_MEMN_GLBAC4	30h

Function

These fully programmable R/W fields define the R/W/X (read/write/execute) access control flags with each of the 4 supported operating modes: SecurePriv, SecureUser, NonsecurePriv, Nonsecure User. The lock bit provides the ability to lock these control access flags and to lock the MBC_BLK_CFG[MBACSEL] field when a locked MBC_MEMN_GLBAC register is selected.

NOTE

The lock field (LK) in MBC0_MEMN_GLBAC1, MBC0_MEMN_GLBAC2, MBC0_MEMN_GLBAC3, MBC0_MEMN_GLBAC6 and MBC0_MEMN_GLBAC7 registers are set to lock state (LK= 1) at reset. Hence these registers cannot be changed and blocks that are assigned to one of these global access control indexes cannot change the access policy index until next reset.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	LK	0														
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	SPR	SPW	SPX	0	SUR	SUW	SUX	0	NPR	NPW	NPX	0	NUR	NUW	NUX
W																
Reset	0	1	0	1	0	1	0	1	0	0	0	0	0	0	0	0

Fields

Field	Function
31 LK	LOCK This 1-bit field provides a mechanism to limit writes to the this register to protect its contents. Once set, this bit remains asserted until the next reset. 0b - This register is not locked and can be altered. 1b - This register is locked and cannot be altered.
30-15 —	Reserved
14	SecurePriv Read

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
SPR	SecurePriv read access control flag 0b - Read access is not allowed in Secure Privilege mode. 1b - Read access is allowed in Secure Privilege mode.
13 SPW	SecurePriv Write SecurePriv write access control flag 0b - Write access is not allowed in Secure Privilege mode. 1b - Write access is allowed in Secure Privilege mode.
12 SPX	SecurePriv Execute SecurePriv execute access control flag 0b - Execute access is not allowed in Secure Privilege mode. 1b - Execute access is allowed in Secure Privilege mode.
11 —	Reserved
10 SUR	SecureUser Read SecureUser read access control flag 0b - Read access is not allowed in Secure User mode. 1b - Read access is allowed in Secure User mode.
9 SUW	SecureUser Write SecureUser write access control flag 0b - Write access is not allowed in Secure User mode. 1b - Write access is allowed in Secure User mode.
8 SUX	SecureUser Execute SecureUser execute access control flag 0b - Execute access is not allowed in Secure User mode. 1b - Execute access is allowed in Secure User mode.
7 —	Reserved
6 NPR	NonsecurePriv Read NonsecurePriv read access control flag 0b - Read access is not allowed in Nonsecure Privilege mode. 1b - Read access is allowed in Nonsecure Privilege mode.

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
5 NPW	NonsecurePriv Write NonsecurePriv write access control flag 0b - Write access is not allowed in Nonsecure Privilege mode. 1b - Write access is allowed in Nonsecure Privilege mode.
4 NPX	NonsecurePriv Execute NonsecurePriv execute access control flag 0b - Execute access is not allowed in Nonsecure Privilege mode. 1b - Execute access is allowed in Nonsecure Privilege mode.
3 —	Reserved
2 NUR	NonsecureUser Read NonsecureUser read access control flag 0b - Read access is not allowed in Nonsecure User mode. 1b - Read access is allowed in Nonsecure User mode.
1 NUW	NonsecureUser Write NonsecureUser write access control flag 0b - Write access is not allowed in Nonsecure User mode. 1b - Write access is allowed in Nonsecure User mode.
0 NUX	NonsecureUser Execute NonsecureUser execute access control flag 0b - Execute access is not allowed in Nonsecure User mode. 1b - Execute access is allowed in Nonsecure User mode.

40.5.1.8 MBC Global Access Control (MBC0_MEMN_GLBAC5)

Offset

Register	Offset
MBC0_MEMN_GLBAC5	34h

Function

These fully programmable R/W fields define the R/W/X (read/write/execute) access control flags with each of the 4 supported operating modes: SecurePriv, SecureUser, NonsecurePriv, Nonsecure User. The lock bit provides the ability to lock these control access flags and to lock the MBC_BLK_CFG[MBACSEL] field when a locked MBC_MEMN_GLBAC register is selected.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

NOTE

The lock field (LK) in MBC0_MEMN_GLBAC1, MBC0_MEMN_GLBAC2, MBC0_MEMN_GLBAC3, MBC0_MEMN_GLBAC6 and MBC0_MEMN_GLBAC7 registers are set to lock state (LK= 1) at reset. Hence these registers cannot be changed and blocks that are assigned to one of these global access control indexes cannot change the access policy index until next reset.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	LK	0														
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	SPR	SPW	SPX	0	SUR	SUW	SUX	0	NPR	NPW	NPX	0	NUR	NUW	NUX
W																
Reset	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0

Fields

Field	Function
31 LK	LOCK This 1-bit field provides a mechanism to limit writes to the this register to protect its contents. Once set, this bit remains asserted until the next reset. 0b - This register is not locked and can be altered. 1b - This register is locked and cannot be altered.
30-15 —	Reserved
14 SPR	SecurePriv Read SecurePriv read access control flag 0b - Read access is not allowed in Secure Privilege mode. 1b - Read access is allowed in Secure Privilege mode.
13 SPW	SecurePriv Write SecurePriv write access control flag 0b - Write access is not allowed in Secure Privilege mode. 1b - Write access is allowed in Secure Privilege mode.
12 SPX	SecurePriv Execute SecurePriv execute access control flag

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	0b - Execute access is not allowed in Secure Privilege mode. 1b - Execute access is allowed in Secure Privilege mode.
11 —	Reserved
10 SUR	SecureUser Read SecureUser read access control flag 0b - Read access is not allowed in Secure User mode. 1b - Read access is allowed in Secure User mode.
9 SUW	SecureUser Write SecureUser write access control flag 0b - Write access is not allowed in Secure User mode. 1b - Write access is allowed in Secure User mode.
8 SUX	SecureUser Execute SecureUser execute access control flag 0b - Execute access is not allowed in Secure User mode. 1b - Execute access is allowed in Secure User mode.
7 —	Reserved
6 NPR	NonsecurePriv Read NonsecurePriv read access control flag 0b - Read access is not allowed in Nonsecure Privilege mode. 1b - Read access is allowed in Nonsecure Privilege mode.
5 NPW	NonsecurePriv Write NonsecurePriv write access control flag 0b - Write access is not allowed in Nonsecure Privilege mode. 1b - Write access is allowed in Nonsecure Privilege mode.
4 NPX	NonsecurePriv Execute NonsecurePriv execute access control flag 0b - Execute access is not allowed in Nonsecure Privilege mode. 1b - Execute access is allowed in Nonsecure Privilege mode.
3	Reserved

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
—	
2 NUR	NonsecureUser Read NonsecureUser read access control flag 0b - Read access is not allowed in Nonsecure User mode. 1b - Read access is allowed in Nonsecure User mode.
1 NUW	NonsecureUser Write NonsecureUser write access control flag 0b - Write access is not allowed in Nonsecure User mode. 1b - Write access is allowed in Nonsecure User mode.
0 NUX	NonsecureUser Execute NonsecureUser execute access control flag 0b - Execute access is not allowed in Nonsecure User mode. 1b - Execute access is allowed in Nonsecure User mode.

40.5.1.9 MBC Global Access Control (MBC0_MEMN_GLBAC6)

Offset

Register	Offset
MBC0_MEMN_GLBAC6	38h

Function

These fully programmable R/W fields define the R/W/X (read/write/execute) access control flags with each of the 4 supported operating modes: SecurePriv, SecureUser, NonsecurePriv, Nonsecure User. The lock bit provides the ability to lock these control access flags and to lock the MBC_BLK_CFG[MBACSEL] field when a locked MBC_MEMN_GLBAC register is selected.

This register is not writable.

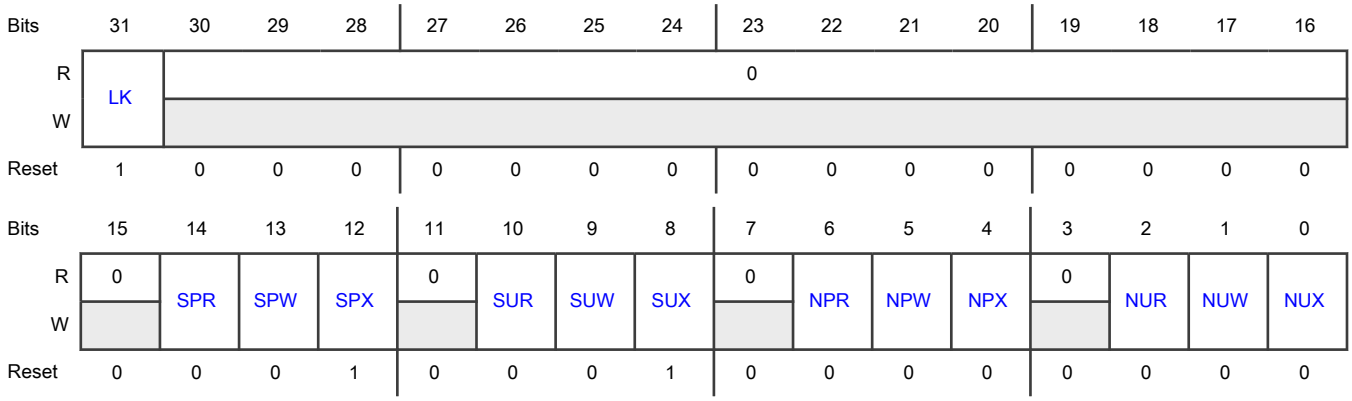
NOTE

The lock field (LK) in MBC0_MEMN_GLBAC1, MBC0_MEMN_GLBAC2, MBC0_MEMN_GLBAC3, MBC0_MEMN_GLBAC6 and MBC0_MEMN_GLBAC7 registers are set to lock state (LK= 1) at reset. Hence these registers cannot be changed and blocks that are assigned to one of these global access control indexes cannot change the access policy index until next reset.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram



Fields

Field	Function
31 LK	LOCK This 1-bit field provides a mechanism to limit writes to the this register to protect its contents. Once set, this bit remains asserted until the next reset. 0b - This register is not locked and can be altered. 1b - This register is locked and cannot be altered.
30-15 —	Reserved
14 SPR	SecurePriv Read SecurePriv read access control flag 0b - Read access is not allowed in Secure Privilege mode. 1b - Read access is allowed in Secure Privilege mode.
13 SPW	SecurePriv Write SecurePriv write access control flag 0b - Write access is not allowed in Secure Privilege mode. 1b - Write access is allowed in Secure Privilege mode.
12 SPX	SecurePriv Execute SecurePriv execute access control flag 0b - Execute access is not allowed in Secure Privilege mode. 1b - Execute access is allowed in Secure Privilege mode.
11 —	Reserved
10	SecureUser Read

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
SUR	SecureUser read access control flag 0b - Read access is not allowed in Secure User mode. 1b - Read access is allowed in Secure User mode.
9 SUW	SecureUser Write SecureUser write access control flag 0b - Write access is not allowed in Secure User mode. 1b - Write access is allowed in Secure User mode.
8 SUX	SecureUser Execute SecureUser execute access control flag 0b - Execute access is not allowed in Secure User mode. 1b - Execute access is allowed in Secure User mode.
7 —	Reserved
6 NPR	NonsecurePriv Read NonsecurePriv read access control flag 0b - Read access is not allowed in Nonsecure Privilege mode. 1b - Read access is allowed in Nonsecure Privilege mode.
5 NPW	NonsecurePriv Write NonsecurePriv write access control flag 0b - Write access is not allowed in Nonsecure Privilege mode. 1b - Write access is allowed in Nonsecure Privilege mode.
4 NPX	NonsecurePriv Execute NonsecurePriv execute access control flag 0b - Execute access is not allowed in Nonsecure Privilege mode. 1b - Execute access is allowed in Nonsecure Privilege mode.
3 —	Reserved
2 NUR	NonsecureUser Read NonsecureUser read access control flag 0b - Read access is not allowed in Nonsecure User mode. 1b - Read access is allowed in Nonsecure User mode.

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
1 NUW	NonsecureUser Write NonsecureUser write access control flag 0b - Write access is not allowed in Nonsecure User mode. 1b - Write access is allowed in Nonsecure User mode.
0 NUX	NonsecureUser Execute NonsecureUser execute access control flag 0b - Execute access is not allowed in Nonsecure User mode. 1b - Execute access is allowed in Nonsecure User mode.

40.5.1.10 MBC Global Access Control (MBC0_MEMN_GLBAC7)

Offset

Register	Offset
MBC0_MEMN_GLBAC7	3Ch

Function

These fully programmable R/W fields define the R/W/X (read/write/execute) access control flags with each of the 4 supported operating modes: SecurePriv, SecureUser, NonsecurePriv, Nonsecure User. The lock bit provides the ability to lock these control access flags and to lock the MBC_BLK_CFG[MBACSEL] field when a locked MBC_MEMN_GLBAC register is selected.

This register is not writable.

NOTE

The lock field (LK) in MBC0_MEMN_GLBAC1, MBC0_MEMN_GLBAC2, MBC0_MEMN_GLBAC3, MBC0_MEMN_GLBAC6 and MBC0_MEMN_GLBAC7 registers are set to lock state (LK= 1) at reset. Hence these registers cannot be changed and blocks that are assigned to one of these global access control indexes cannot change the access policy index until next reset.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	LK	0														
W																
Reset	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	SPR	SPW	SPX	0	SUR	SUW	SUX	0	NPR	NPW	NPX	0	NUR	NUW	NUX
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Fields

Field	Function
31 LK	LOCK This 1-bit field provides a mechanism to limit writes to the this register to protect its contents. Once set, this bit remains asserted until the next reset. 0b - This register is not locked and can be altered. 1b - This register is locked and cannot be altered.
30-15 —	Reserved
14 SPR	SecurePriv Read SecurePriv read access control flag 0b - Read access is not allowed in Secure Privilege mode. 1b - Read access is allowed in Secure Privilege mode.
13 SPW	SecurePriv Write SecurePriv write access control flag 0b - Write access is not allowed in Secure Privilege mode. 1b - Write access is allowed in Secure Privilege mode.
12 SPX	SecurePriv Execute SecurePriv execute access control flag 0b - Execute access is not allowed in Secure Privilege mode. 1b - Execute access is allowed in Secure Privilege mode.
11 —	Reserved
10 SUR	SecureUser Read SecureUser read access control flag 0b - Read access is not allowed in Secure User mode. 1b - Read access is allowed in Secure User mode.
9 SUW	SecureUser Write SecureUser write access control flag 0b - Write access is not allowed in Secure User mode. 1b - Write access is allowed in Secure User mode.
8 SUX	SecureUser Execute SecureUser execute access control flag

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	0b - Execute access is not allowed in Secure User mode. 1b - Execute access is allowed in Secure User mode.
7 —	Reserved
6 NPR	NonsecurePriv Read NonsecurePriv read access control flag 0b - Read access is not allowed in Nonsecure Privilege mode. 1b - Read access is allowed in Nonsecure Privilege mode.
5 NPW	NonsecurePriv Write NonsecurePriv write access control flag 0b - Write access is not allowed in Nonsecure Privilege mode. 1b - Write access is allowed in Nonsecure Privilege mode.
4 NPX	NonsecurePriv Execute NonsecurePriv execute access control flag 0b - Execute access is not allowed in Nonsecure Privilege mode. 1b - Execute access is allowed in Nonsecure Privilege mode.
3 —	Reserved
2 NUR	NonsecureUser Read NonsecureUser read access control flag 0b - Read access is not allowed in Nonsecure User mode. 1b - Read access is allowed in Nonsecure User mode.
1 NUW	NonsecureUser Write NonsecureUser write access control flag 0b - Write access is not allowed in Nonsecure User mode. 1b - Write access is allowed in Nonsecure User mode.
0 NUX	NonsecureUser Execute NonsecureUser execute access control flag 0b - Execute access is not allowed in Nonsecure User mode. 1b - Execute access is allowed in Nonsecure User mode.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

40.5.1.11 MBC Memory Block Configuration Word (MBC0_DOM0_MEM0_BLK_CFG_W0 - MBC0_DOM0_MEM2_BLK_CFG_W0)

Offset

Register	Offset
MBC0_DOM0_MEM0_BLK_CFG_W0	40h
MBC0_DOM0_MEM0_BLK_CFG_W1	44h
MBC0_DOM0_MEM1_BLK_CFG_W0	180h
MBC0_DOM0_MEM1_BLK_CFG_W1	184h
MBC0_DOM0_MEM2_BLK_CFG_W0	1A8h

Function

MBC[m]_DOM[d]_MEM[s]_BLK_CFG_W[w], where,

- m - mbc index
- d - domain index
- s - memory target index
- w - word index

These registers are read/write for both the alternate view of the NSE bit and the 3-bit MBACSEL field. This is an array of 4 bit fields defining the block configuration for the given submemory. Each 4-bit field includes an alternative view of the associated NSE bit plus a 3-bit access control select that selects the global access control value that applies to the referenced memory block.

For a given memory block, B, if the value programmed in the MBACSELn field selects a locked MBC_MEMN_GLBACr register, the MBACSEL field is locked until the next reset. Depending on BLK_CFG_W, the NSEn/MBASELn fields correspond to different blocks. The following table describes the relationship between W (word index) and B (block number).

MEM0 supports up to 512 blocks ; W0 - W63

MEM1-3 supports up to 64 block ; W0 - W7

NOTE

This register is shown with fields such that it covers entire 32 bit register. However, actual fields may be less if number of blocks are such that fields don't align to 32 bits. These absent fields may be shown but user can refer to table below to deduce actual fields and treat absent fields as reserved.

Table 244. Block Config Word to Block Number Relationship

Block Config Word	NSE7/ MBACSEL7	NSE6/ MBACSEL6	NSE5/ MBACSEL5	NSE4/ MBACSEL4	NSE3/ MBACSEL3	NSE2/ MBACSEL2	NSE1/ MBACSEL1	NSE0/ MBACSEL0
W0	block 7	block 6	block 5	block 4	block 3	block 2	block 1	block 0

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table 244. Block Config Word to Block Number Relationship (continued)

Block Config Word	NSE7/ MBACSEL7	NSE6/ MBACSEL6	NSE5/ MBACSEL5	NSE4/ MBACSEL4	NSE3/ MBACSEL3	NSE2/ MBACSEL2	NSE1/ MBACSEL1	NSE0/ MBACSEL0
W1	block 15	block 14	block 13	block 12	block 11	block 10	block 9	block 8
W2	block 23	block 22	block 21	block 20	block 19	block 18	block 17	block 16
W3	block 31	block 30	block 29	block 28	block 27	block 26	block 25	block 24
W4	block 39	block 38	block 37	block 36	block 35	block 34	block 33	block 32
W5	block 47	block 46	block 45	block 44	block 43	block 42	block 41	block 40
W6	block 55	block 54	block 53	block 52	block 51	block 50	block 49	block 48
W7	block 63	block 62	block 61	block 60	block 59	block 58	block 57	block 56
For MEM0, block 64 - block 511								
W8	block 71	block 70	block 69	block 68	block 67	block 66	block 65	block 64
. . .								
W63	block 511	block 510	block 509	block 508	block 507	block 506	block 505	block 504

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	NSE7	MBACSEL7			NSE6	MBACSEL6			NSE5	MBACSEL5			NSE4	MBACSEL4		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	NSE3	MBACSEL3			NSE2	MBACSEL2			NSE1	MBACSEL1			NSE0	MBACSEL0		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31 NSE7	NonSecure Enable for block B

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	0b - Secure accesses to block B are based on corresponding MBACSEL field in this register (MBCm_DOMd_MEMs_BLK_CFG_Ww[MBACSEL]), nonsecure accesses to block B are not allowed. 1b - Secure executions to block B are not allowed. Secure read, secure write and nonsecure accesses to block B are based on corresponding MBACSEL field in this register (MBCm_DOMd_MEMs_BLK_CFG_Ww[MBACSEL]).
30-28 MBACSEL7	Memory Block Access Control Select for block B This field selects the global access control associated with block B. **NOTE** This field is locked (read-only) if MBACSEL7 = r and MBC_MEMN_GLBACr[LK] = 1. 000b - select MBC_MEMN_GLBAC0 access control policy for block B 001b - select MBC_MEMN_GLBAC1 access control policy for block B 010b - select MBC_MEMN_GLBAC2 access control policy for block B 011b - select MBC_MEMN_GLBAC3 access control policy for block B 100b - select MBC_MEMN_GLBAC4 access control policy for block B 101b - select MBC_MEMN_GLBAC5 access control policy for block B 110b - select MBC_MEMN_GLBAC6 access control policy for block B 111b - select MBC_MEMN_GLBAC7 access control policy for block B
27 NSE6	NonSecure Enable for block B 0b - Secure accesses to block B are based on corresponding MBACSEL field in this register (MBCm_DOMd_MEMs_BLK_CFG_Ww[MBACSEL]), nonsecure accesses to block B are not allowed. 1b - Secure executions to block B are not allowed. Secure read, secure write and nonsecure accesses to block B are based on corresponding MBACSEL field in this register (MBCm_DOMd_MEMs_BLK_CFG_Ww[MBACSEL]).
26-24 MBACSEL6	Memory Block Access Control Select for block B This field selects the global access control associated with block B. **NOTE** This field is locked (read-only) if MBACSEL6 = r and MBC_MEMN_GLBACr[LK] = 1. 000b - select MBC_MEMN_GLBAC0 access control policy for block B 001b - select MBC_MEMN_GLBAC1 access control policy for block B 010b - select MBC_MEMN_GLBAC2 access control policy for block B 011b - select MBC_MEMN_GLBAC3 access control policy for block B 100b - select MBC_MEMN_GLBAC4 access control policy for block B

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	101b - select MBC_MEMN_GLBAC5 access control policy for block B 110b - select MBC_MEMN_GLBAC6 access control policy for block B 111b - select MBC_MEMN_GLBAC7 access control policy for block B
23 NSE5	NonSecure Enable for block B 0b - Secure accesses to block B are based on corresponding MBACSEL field in this register (MBCm_DOMd_MEMs_BLK_CFG_Ww[MBACSEL]), nonsecure accesses to block B are not allowed. 1b - Secure executions to block B are not allowed. Secure read, secure write and nonsecure accesses to block B are based on corresponding MBACSEL field in this register (MBCm_DOMd_MEMs_BLK_CFG_Ww[MBACSEL]).
22-20 MBACSEL5	Memory Block Access Control Select for block B This field selects the global access control associated with block B. **NOTE** This field is locked (read-only) if MBACSEL5 = r and MBC_MEMN_GLBACr[LK] = 1. 000b - select MBC_MEMN_GLBAC0 access control policy for block B 001b - select MBC_MEMN_GLBAC1 access control policy for block B 010b - select MBC_MEMN_GLBAC2 access control policy for block B 011b - select MBC_MEMN_GLBAC3 access control policy for block B 100b - select MBC_MEMN_GLBAC4 access control policy for block B 101b - select MBC_MEMN_GLBAC5 access control policy for block B 110b - select MBC_MEMN_GLBAC6 access control policy for block B 111b - select MBC_MEMN_GLBAC7 access control policy for block B
19 NSE4	NonSecure Enable for block B 0b - Secure accesses to block B are based on corresponding MBACSEL field in this register (MBCm_DOMd_MEMs_BLK_CFG_Ww[MBACSEL]), nonsecure accesses to block B are not allowed. 1b - Secure executions to block B are not allowed. Secure read, secure write and nonsecure accesses to block B are based on corresponding MBACSEL field in this register (MBCm_DOMd_MEMs_BLK_CFG_Ww[MBACSEL]).
18-16 MBACSEL4	Memory Block Access Control Select for block B This field selects the global access control associated with block B. **NOTE** This field is locked (read-only) if MBACSEL4 = r and MBC_MEMN_GLBACr[LK] = 1. 000b - select MBC_MEMN_GLBAC0 access control policy for block B

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	001b - select MBC_MEMN_GLBAC1 access control policy for block B 010b - select MBC_MEMN_GLBAC2 access control policy for block B 011b - select MBC_MEMN_GLBAC3 access control policy for block B 100b - select MBC_MEMN_GLBAC4 access control policy for block B 101b - select MBC_MEMN_GLBAC5 access control policy for block B 110b - select MBC_MEMN_GLBAC6 access control policy for block B 111b - select MBC_MEMN_GLBAC7 access control policy for block B
15 NSE3	NonSecure Enable for block B 0b - Secure accesses to block B are based on corresponding MBACSEL field in this register (MBCm_DOMd_MEMs_BLK_CFG_Ww[MBACSEL]), nonsecure accesses to block B are not allowed. 1b - Secure executions to block B are not allowed. Secure read, secure write and nonsecure accesses to block B are based on corresponding MBACSEL field in this register (MBCm_DOMd_MEMs_BLK_CFG_Ww[MBACSEL]).
14-12 MBACSEL3	Memory Block Access Control Select for block B This field selects the global access control associated with block B. **NOTE** This field is locked (read-only) if MBACSEL3 = r and MBC_MEMN_GLBACr[LK] = 1. 000b - select MBC_MEMN_GLBAC0 access control policy for block B 001b - select MBC_MEMN_GLBAC1 access control policy for block B 010b - select MBC_MEMN_GLBAC2 access control policy for block B 011b - select MBC_MEMN_GLBAC3 access control policy for block B 100b - select MBC_MEMN_GLBAC4 access control policy for block B 101b - select MBC_MEMN_GLBAC5 access control policy for block B 110b - select MBC_MEMN_GLBAC6 access control policy for block B 111b - select MBC_MEMN_GLBAC7 access control policy for block B
11 NSE2	NonSecure Enable for block B 0b - Secure accesses to block B are based on corresponding MBACSEL field in this register (MBCm_DOMd_MEMs_BLK_CFG_Ww[MBACSEL]), nonsecure accesses to block B are not allowed. 1b - Secure executions to block B are not allowed. Secure read, secure write and nonsecure accesses to block B are based on corresponding MBACSEL field in this register (MBCm_DOMd_MEMs_BLK_CFG_Ww[MBACSEL]).
10-8 MBACSEL2	Memory Block Access Control Select for block B This field selects the global access control associated with block B.

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	NOTE This field is locked (read-only) if MBACSEL2 = r and MBC_MEMN_GLBACr[LK] = 1. 000b - select MBC_MEMN_GLBAC0 access control policy for block B 001b - select MBC_MEMN_GLBAC1 access control policy for block B 010b - select MBC_MEMN_GLBAC2 access control policy for block B 011b - select MBC_MEMN_GLBAC3 access control policy for block B 100b - select MBC_MEMN_GLBAC4 access control policy for block B 101b - select MBC_MEMN_GLBAC5 access control policy for block B 110b - select MBC_MEMN_GLBAC6 access control policy for block B 111b - select MBC_MEMN_GLBAC7 access control policy for block B
7 NSE1	NonSecure Enable for block B 0b - Secure accesses to block B are based on corresponding MBACSEL field in this register (MBCm_DOMd_MEMs_BLK_CFG_Ww[MBACSEL]), nonsecure accesses to block B are not allowed. 1b - Secure executions to block B are not allowed. Secure read, secure write and nonsecure accesses to block B are based on corresponding MBACSEL field in this register (MBCm_DOMd_MEMs_BLK_CFG_Ww[MBACSEL]).
6-4 MBACSEL1	Memory Block Access Control Select for block B This field selects the global access control associated with block B. **NOTE** This field is locked (read-only) if MBACSEL1 = r and MBC_MEMN_GLBACr[LK] = 1. 000b - select MBC_MEMN_GLBAC0 access control policy for block B 001b - select MBC_MEMN_GLBAC1 access control policy for block B 010b - select MBC_MEMN_GLBAC2 access control policy for block B 011b - select MBC_MEMN_GLBAC3 access control policy for block B 100b - select MBC_MEMN_GLBAC4 access control policy for block B 101b - select MBC_MEMN_GLBAC5 access control policy for block B 110b - select MBC_MEMN_GLBAC6 access control policy for block B 111b - select MBC_MEMN_GLBAC7 access control policy for block B
3 NSE0	NonSecure Enable for block B 0b - Secure accesses to block B are based on corresponding MBACSEL field in this register (MBCm_DOMd_MEMs_BLK_CFG_Ww[MBACSEL]), nonsecure accesses to block B are not allowed.

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	1b - Secure executions to block B are not allowed. Secure read, secure write and nonsecure accesses to block B are based on corresponding MBACSEL field in this register (MBCm_DOMd_MEMs_BLK_CFG_Ww[MBACSEL]).
2-0 MBACSEL0	Memory Block Access Control Select for block B This field selects the global access control associated with block B. **NOTE** This field is locked (read-only) if MBACSEL0 = r and MBC_MEMN_GLBACr[LK] = 1. 000b - select MBC_MEMN_GLBAC0 access control policy for block B 001b - select MBC_MEMN_GLBAC1 access control policy for block B 010b - select MBC_MEMN_GLBAC2 access control policy for block B 011b - select MBC_MEMN_GLBAC3 access control policy for block B 100b - select MBC_MEMN_GLBAC4 access control policy for block B 101b - select MBC_MEMN_GLBAC5 access control policy for block B 110b - select MBC_MEMN_GLBAC6 access control policy for block B 111b - select MBC_MEMN_GLBAC7 access control policy for block B

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Chapter 41
True Random Generator (TRNG)

41.1 Chip-specific TRNG information

Table 245. Reference links to related information

Topic	Related module	Reference
Full description	TRNG	TRNG
System memory map		Memory map
Clocking		Clock distribution
System debug		Debug
Signal multiplexing	Port control	Signal multiplexing

41.1.1 Module instance

This chip has one instance of the TRNG module.

41.1.2 Security considerations

For secure applications, NXP recommends that this module is configured for secure and privileged access only to prevent unintentional or malicious modification of the system operation by nonsecure software.

41.2 Overview

The True Random Number Generator (TRNG) is a hardware accelerator module that generates 256-bit entropy as needed by an entropy-consuming module or by other post-processing functions. An appropriate entropy consumer is a deterministic random bit generator (DRBG) that can be implemented to achieve both randomness and cryptographic-strength using the TRNG output as its seed. The DRBG is not part of this module.

The entropy generated by a TRNG is intended to be used by functions (for example, to seed DRBG), whose output then can be used by functions that generate secret keys, per-message secrets, random challenges, and other similar quantities used in cryptographic algorithms. In each of these cases, it is important that a random number be difficult to guess or predict. It is important that a random number is at least as difficult to predict as it is difficult to break the cryptographic algorithm with which it is being used. This stringent requirement is particularly difficult to fulfill if the entropy source from a TRNG contains bias and/or correlation. To increase the trustworthiness and quality of the generated random data, DRBGs are often used to post-process the output of a TRNG.

This document describes the TRNG design functionality and usage.

Note that before entropy can be obtained from the TRNG, it must be initialized by setting the appropriate TRNG registers.

41.2.1 Block Diagram

The TRNG contains the following sub-modules: register interface, counters, config registers, health tests, and free-running oscillators (OSC).

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

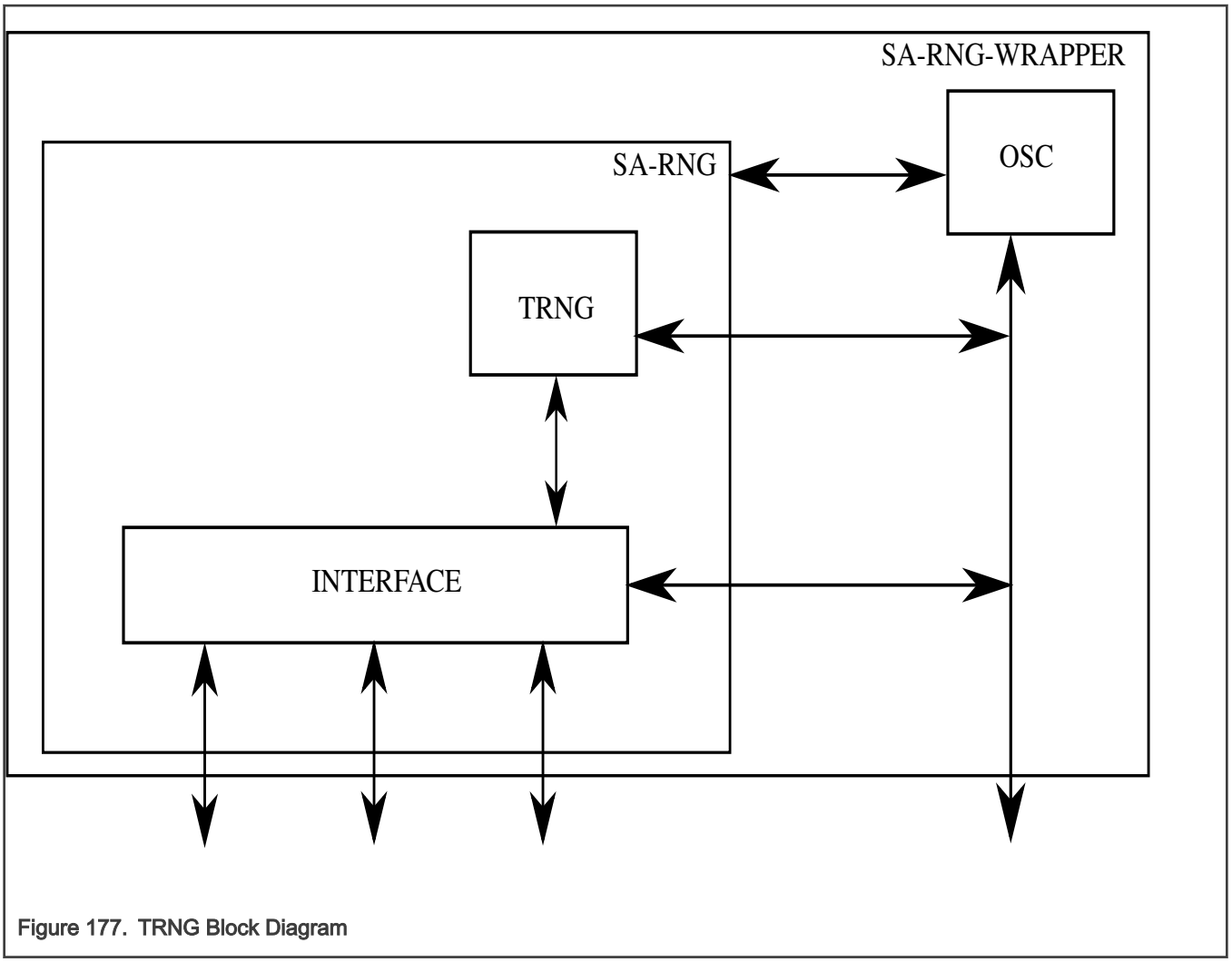


Figure 177. TRNG Block Diagram

41.3 Functional Description

The TRNG consists of several functional sub-modules. Its overall functionality can be described from the top-level in terms of generating entropy for seed generation.

TRNG generates random bits by collecting bits from a random noise source. This noise causes various small changes in the period of the oscillator. Therefore, if the count of the ring oscillator clock cycles is sampled after a known period of time, this count will vary each time the sample is taken. By using the variation in this count over a large number of samples, random bits can be derived. In addition to generating entropy, the TRNG also performs several statistical tests on its output.

41.4 Register Interface Usage

The TRNG register interface is used to read and write registers within TRNG for the following purposes:

- to configure TRNG
- to read the generated entropy

Note that accesses to registers must be 32-bit reads or writes.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

41.5 Operation

The TRNG is designed to operate as a peripheral and can be controlled via the register interface. In order to write to many TRNG registers, the MCTL register must be set to programming mode (TRNG_MCTL[PRGM] = 1). After a power-on reset (POR) or an asynchronous system hard reset, the TRNG resets to programming mode. TRNG will not generate entropy until it has transitioned from programming mode to run mode.

There are multiple operations that the user (programmer/integrator) will want to do with a TRNG.

1. Initialize.

   Set up the parameters to proper values and then start generation of the first block of entropy. This is done after reset or any time the user wants to update operating parameters and/or built-in self test parameter bounds. Part of this step involves clearing of registers FRQCNT and OSC2_FRQCNT by performing a read to their corresponding address location, prior to transitioning TRNG programming mode into run mode.

   For more details, see Standalone Operation Programming Flow

2. Read entropy and generate next block of entropy.

   This is the normal flow of operation and is looped until enough entropy has been generated.

41.5.1 Standalone Operation Programming Flow

After reset, the TRNG will be in programming mode with the ok-to-stop bit asserted (MCTL[TSTOP_OK] = 1). Below is an example of a normal initialization flow.

1. Initialize TRNG:
   a. Write the desired sample size and entropy delay values to the SDCTL register.
   b. Program TRNG's self-test bounds registers. Refer to Sample Size Adjustment Programming Flow for recommended values based on sample size.
   c. Set the MCTL[TRNG_ACC] bit to enable access to the entropy registers and other critical TRNG registers.
   d. Optionally, clear the FRQCNT and OSC2_FRQCNT registers by reading them.
   e. Optionally, enable interrupts in the INT_MASK register.
2. Move TRNG into run mode:
   a. Clear MCTL[PRGM] to remove the TRNG from programming mode. This will also start entropy generation.
3. Poll TRNG (MCTL) to check if valid entropy was generated or error occurred:
   a. Wait for valid entropy by polling the MCTL[TSTOP_OK] bit or waiting for a TRNG interrupt.
   b. Check whether bit MCTL[ENT_VAL] is set to indicate that the entropy is valid (no errors), or else, whether bit MCTL[ERR] is set to indicate error has occurred during entropy generation.
   c. Read the 256-bit entropy from ENT0 to ENT7.

   NOTE

   Reading ENT7 will reset the entropy, so it should always be read last.

   d. Repeat this process as needed to read additional entropy values.

41.5.2 Interrupt Programming Flow

The TRNG has four interrupts that are managed using three TRNG registers: INT_CTRL, INT_STATUS and INT_MASK registers. The interrupts are INTG_ERR, FRQ_CT_FAIL, ENT_VAL and HW_ERR. By default, these interrupts are disabled. They have to be individually enabled, as needed, to work.

For example, to enable the ENT_VAL interrupt, the corresponding bit in the INT_MASK register must be set (INT_MASK[ENT_VAL] = 1). When a new interrupt is triggered, the corresponding bit in the INT_STATUS register will assert

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

(INT_STATUS[ENT_VAL]). To clear an asserted interrupt, the corresponding bit in the INT_CTRL register is written with value 0 (INT_CTRL[ENT_VAL] = 0).

After a power cycle or reset, the TRNG interrupts need to be enabled again. Note that these interrupt bits correspond to the MCTL[INTG_ERR], MCTL[FCT_FAIL], MCTL[ENT_VAL] and MCTL[ERR]. These MCTL bits will always assert/de-assert regardless of whether interrupts are masked or temporarily enabled/disabled. The interrupt registers mirror these MCTL bits but are independent.

Below is an example of an interrupt setup and programming flow for ENT_VAL, HW_ERR, and FRQ_CT_FAIL interrupts can be enabled in the same way.

1. Initialize TRNG:
    a. Write the desired sample size and entropy delay values to the SDCTL register.
    b. Program TRNG's self-test bounds registers. Refer to Sample Size Adjustment Programming Flow for recommended values based on sample size.
    c. Set the MCTL[TRNG_ACC] bit to enable access to the entropy registers and other critical TRNG registers.
    d. Optionally, clear the FRQCNT and OSC2_FRQCNT registers by reading them.
    e. Optionally, enable interrupts in the INT_MASK register
    f. Turn on the ENT_VAL interrupt functionality by setting INT_CTRL[ENT_VAL] = 1 and INT_MASK[ENT_VAL] = 1.
2. Move TRNG into run mode:
    a. Clear MCTL[PRGM] to remove the TRNG from programming mode. This will also start entropy generation.
3. Wait for an interrupt from TRNG and check whether a valid entropy was generated or error occurred:
    a. When entropy is ready, the active-low interrupt is expected to trigger. This should also be reflected in the interrupt status register (INT_STATUS[ENT_VAL] = 1).
    b. Check whether bit MCTL[ENT_VAL] is set to indicate that the entropy is valid (no errors), or else, whether bit MCTL[ERR] is set to indicate error has occurred during entropy generation.
    c. Read the 256-bit entropy from ENT0 to ENT7.

NOTE

Reading ENT7 will reset the entropy, so it should always be read last.

    d. Reading the ENT7 register will clear the ENT_VAL interrupt. This should also be reflected in the interrupt status register (INT_STATUS[ENT_VAL] = 0).
    e. Repeat this process as needed to read additional entropy values.

41.5.3 Sample Size Adjustment Programming Flow

The sample size (SDCTL[SAMP_SIZE]) parameter can be adjusted to speed up or to limit the number of bits that an application wants to use. The greater the sample size, the more sensitive the self tests will be to entropy loss. However, this slows down entropy generation. The tradeoff between the speed versus the quality of entropy needs to be balanced per application.

Every time the sample size parameter is updated, the bounds of the built-in self tests (BIST) must also be updated to match the acceptable error ranges for the new sample size.

Below are the empirical options for some sample size parameters and their corresponding self bounds adjustment, mentioned in both table and pseudo-code formats.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table 246. Bound values for the BIST For Some Sample Sizes

Configuration Register	SDCTL[SAMP_SIZE] Write Value		
	128	256	512
SCMISC	0x0001001D	0x0001001F	0x00010020
SCML	0x003D005E	0x005600AB	0x007A013D
SCR1L	0x00270027	0x0038003F	0x0050006B
SCR2L	0x00190018	0x00260026	0x0037003E
SCR3L	0x00120011	0x001A0019	0x00270027

Pseudo-code: Programming values for the BIST for some sample sizes

```
// example Seed Control Register (SDCTL) parameters
samp_size    = 128;  // decimal

// target rejection ratio = 1.00E-06

if ( samp_size == 128 ) {
    reg32_write( TRNG_SCMISC, 0x0001001D ); // retry once, long run max 29

                                            // min range max
    reg32_write( TRNG_SCML,   0x003D005E ); //  33    61  94        monobit limit]
    reg32_write( TRNG_SCR1L,  0x00270027 ); //   0    39  39 [run length 1  limit]
    reg32_write( TRNG_SCR2L,  0x00190018 ); //  -1    25  24 [run length 2  limit]
    reg32_write( TRNG_SCR3L,  0x00120011 ); //  -1    18  17 [run length 3  limit]

} else if ( samp_size == 256 ) {
    reg32_write( TRNG_SCMISC, 0x0001001F ); // retry once, long run max 31

                                            // min range max
    reg32_write( TRNG_SCML,   0x005600AB ); //  85    86 171 [     monobit  limit]
    reg32_write( TRNG_SCR1L,  0x0038003f ); //   7    56  63 [run length 1  limit]
    reg32_write( TRNG_SCR2L,  0x00260026 ); //   0    38  38 [run length 2  limit]
    reg32_write( TRNG_SCR3L,  0x001A0019 ); //  -1    26  25 [run length 3  limit]


} else if ( samp_size == 512 ) {
    reg32_write( TRNG_SCMISC, 0x00010020 ); // retry once, long run max 32

                                            //  min range  max
    reg32_write( TRNG_SCML,   0x007A013D ); //  195   122  317 [     monobit  limit]
    reg32_write( TRNG_SCR1L,  0x0050006B ); //   27    80  107 [run length 1  limit]
    reg32_write( TRNG_SCR2L,  0x0037003E ); //    7    55   62 [run length 2  limit]
    reg32_write( TRNG_SCR3L,  0x00270027 ); //    0    39   39 [run length 3  limit]

}

// adjust delay
reg32_write( TRNG_SDCTL, ( (reg32_read( TRNG_SDCTL ) & 0xFFFF0000) | (uint32_t)samp_size ));
```

After the sample size parameter has been set, entropy can be requested normally without generating any errors.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

41.6 Other Applications

The TRNG can also be used by a post-processing pseudo-random number generator function. For example, TRNG can be used to seed a hardware- or software-based implementation of a deterministic random bit generator (DRBG) as defined by SP800-90A.

41.7 Memory Map and Registers

41.7.1 TRNG register descriptions

All accesses of undefined addresses always return zero when read and generate a bus error when written. Writes to undefined and read-only addresses are ignored. Undefined addresses are those undocumented, protected or reserved addresses within the range of the addresses defined in the memory map below. The register addresses shown in the Memory Map below represent how these registers are accessed over the register bus as 32-bit words. Accesses to registers must use full-word (32-bit) reads or writes.

The format and fields in each register are defined below. Some of the register format figures apply to several different registers. In such cases a different register name will be associated with each of the register offset addresses that appear at the top of the register format figure. Although these registers share the same format, they are independent registers.

41.7.1.1 TRNG memory map

TRNG0 base address: 400E_C000h

Offset	Register	Width (In bits)	Access	Reset value
0h	Miscellaneous Control Register (MCTL)	32	RW	0001_2000h
4h	Statistical Check Miscellaneous Register (SCMISC)	32	RW	0001_001Fh
8h	Poker Range Register (PKRRNG)	32	RW	0000_023Ah
Ch	Poker Maximum Limit Register (PKRMAX)	32	RW	0000_0640h
Ch	Poker Square Calculation Result Register (PKRSQ)	32	R	0000_0000h
10h	Seed Control Register (SDCTL)	32	RW	0C80_0200h
14h	Sparse Bit Limit Register (SBLIM)	32	RW	0000_003Fh
14h	Total Samples Register (TOTSAM)	32	R	0000_0000h
18h	Frequency Count Minimum Limit Register (FRQMIN)	32	RW	0000_0640h
18h	Oscillator-2 Frequency Count Register (OSC2_FRQCNT)	32	R	0000_0000h
1Ch	Frequency Count Register (FRQCNT)	32	R	0000_0000h
1Ch	Frequency Count Maximum Limit Register (FRQMAX)	32	RW	0000_6400h
20h	Statistical Check Monobit Count Register (SCMC)	32	R	0000_0000h
20h	Statistical Check Monobit Limit Register (SCML)	32	RW	0078_013Ch
24h	Statistical Check Run Length 1 Count Register (SCR1C)	32	R	0000_0000h
24h	Statistical Check Run Length 1 Limit Register (SCR1L)	32	RW	004F_006Bh
28h	Statistical Check Run Length 2 Count Register (SCR2C)	32	R	0000_0000h
28h	Statistical Check Run Length 2 Limit Register (SCR2L)	32	RW	0036_003Eh

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Offset	Register	Width (In bits)	Access	Reset value
2Ch	Statistical Check Run Length 3 Count Register (SCR3C)	32	R	0000_0000h
2Ch	Statistical Check Run Length 3 Limit Register (SCR3L)	32	RW	002D_0037h
30h	Statistical Check Run Length 4 Count Register (SCR4C)	32	R	0000_0000h
30h	Statistical Check Run Length 4 Limit Register (SCR4L)	32	RW	001B_001Ah
34h	Statistical Check Run Length 5 Count Register (SCR5C)	32	R	0000_0000h
34h	Statistical Check Run Length 5 Limit Register (SCR5L)	32	RW	0013_0012h
38h	Statistical Check Run Length 6+ Count Register (SCR6PC)	32	R	0000_0000h
38h	Statistical Check Run Length 6+ Limit Register (SCR6PL)	32	RW	0012_0011h
3Ch	Status Register (STATUS)	32	R	0000_0000h
40h - 5Ch	Entropy Read Register (ENT0 - ENT7)	32	R	0000_0000h
80h	Statistical Check Poker Count 1 and 0 Register (PKRCNT10)	32	R	0000_0000h
84h	Statistical Check Poker Count 3 and 2 Register (PKRCNT32)	32	R	0000_0000h
88h	Statistical Check Poker Count 5 and 4 Register (PKRCNT54)	32	R	0000_0000h
8Ch	Statistical Check Poker Count 7 and 6 Register (PKRCNT76)	32	R	0000_0000h
90h	Statistical Check Poker Count 9 and 8 Register (PKRCNT98)	32	R	0000_0000h
94h	Statistical Check Poker Count B and A Register (PKRCNTBA)	32	R	0000_0000h
98h	Statistical Check Poker Count D and C Register (PKRCNTDC)	32	R	0000_0000h
9Ch	Statistical Check Poker Count F and E Register (PKRCNTFE)	32	R	0000_0000h
A0h	Security Configuration Register (SEC_CFG)	32	RW	0000_0000h
A4h	Interrupt Control Register (INT_CTRL)	32	RW	See section
A8h	Mask Register (INT_MASK)	32	RW	0000_0000h
ACh	Interrupt Status Register (INT_STATUS)	32	R	0000_0000h
ECh	TRNG Oscillator 2 Control Register (OSC2_CTL)	32	RW	0000_1000h
F0h	Version ID Register (MS) (VID1)	32	R	0030_140Ch
F4h	Version ID Register (LS) (VID2)	32	R	0C0A_0000h

41.7.1.2 Miscellaneous Control Register (MCTL)

Offset

Register	Offset
MCTL	0h

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Function

This register is intended to be used for programming, configuring, and testing the TRNG. It is the main register to read/write in order to enable entropy generation, to stop entropy generation, and to block access to entropy registers. This is done via the TRNG_ACC and PRGM bit below.

The Miscellaneous Control Register is a read/write register used to control the True Random Number Generator (TRNG) access, operation and test.

NOTE

Note that in many cases two TRNG registers share the same address, and a particular register at the shared address is selected based upon the value in the PRGM field of the MCTL register.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	INTG_ERR	Reserved														PRGM
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	OSC2_FA...	Reserved	TSTOP_OK	ERR	Reserved	ENT_VAL	FCT_VAL	FCT_FAIL	Reserved		TRNG_ACC		OSC_DIV		Reserved	
W				W1C						RST_DEF		DIS_SLF...				
Reset	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31 INTG_ERR	Integrity Error An internal fault has been detected. 0b - TRNG detected no internal bit error 1b - TRNG detected internal bit error(s)
30-17 —	Reserved
16 PRGM	Program Mode No Entropy value will be generated while the TRNG is in Program Mode. Note that different TRNG registers are accessible at the same address depending on whether PRGM is set to 1 or 0. This is noted in the TRNG register descriptions. **NOTE** If PRGM is set to 1, then TRNG_ACC should also be set to 1. 0b - TRNG is in Run Mode

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	1b - TRNG is in Program Mode
15 OSC2_FAIL	Oscillator 2 Failure OSC2_FAIL. 0b - Oscillator 2 is running. 1b - Oscillator 2 has failed (see OSC2_CTL[OSC_FAILSAFE_LMT]).
14 —	Reserved
13 TSTOP_OK	TRNG is ok to stop Software can check that this bit is a 1 before transitioning pd_main.trng0 to low power mode (pd_main.trng0 clock stopped). pd_main.trng0 turns on the TRNG free-running ring oscillator whenever new entropy is being generated and turns off the ring oscillator when entropy generation is complete. If the pd_main.trng0 clock is stopped while the TRNG ring oscillator is running, the oscillator will continue running even though the pd_main.trng0 clock is stopped. TSTOP_OK is set when the TRNG ring oscillator is not running. Setting the MCTL[PRGM] bit will also cause TSTOP_OK to assert. 0b - TRNG is generating entropy and is not ok to stop 1b - TRNG is not generating entropy and is ok to stop
12 ERR	Error Status Read: Error status. Write: Write 1 to clear errors. Writing 0 has no effect. 0b - No error 1b - Error detected
11 —	Reserved
10 ENT_VAL	Entropy Valid Will set after an entropy value is generated and only if TRNG_ACC bit is set. Will be cleared at most one (1) bus clock cycle after reading the ENT7 register. (ENT0 through ENT6 should be read before reading ENT7). 0b - Entropy is not valid 1b - Entropy is valid
9 FCT_VAL	Frequency Count Valid Indicates that a valid frequency count may be read from FRQCNT. 0b - Frequency Count is not valid 1b - Frequency Count is valid
8	Frequency Count Fail

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
FCT_FAIL	The frequency counter has detected a failure. This may be due to improper programming of the FRQMAX and/or FRQMIN registers, or a hardware failure in the ring oscillator. This error may be cleared by writing a 1 to the ERR bit. See OSC2_FAIL to determine if the failure is due to oscillator 1 or oscillator 2. • In Single-Osc1 mode (OSC2_CTL[TRNG_ENT_CTL] = 00b): — if FCT_FAIL = 1, oscillator 1 has failed • In Dual-Osc mode (OSC2_CTL[TRNG_ENT_CTL] = 01b): — if FCT_FAIL = 1, oscillator 1 has failed — if OSC2_FAIL = 1, oscillator 2 has failed and in addition, FCT_FAIL also gets set • In Single-Osc2 mode (OSC2_CTL[TRNG_ENT_CTL] = 10b): — if FCT_FAIL = 1, oscillator 2 has failed
7 —	Reserved
6 RST_DEF	Reset Defaults This bit is writable only if PRGM bit is 1, or, PRGM bit is being written to 1 simultaneously while writing to this bit. This bit has higher priority if both this and MCTL[DIS_SLF_TST] bits are written simultaneously. Reading this bit always produces a 0. This is a self-clearing bit. 0b - No impact. 1b - Writing a 1 to this bit clears various TRNG registers, and bits within registers, to their default state.
5 TRNG_ACC	TRNG Access Mode If this bit is set to 1, the TRNG will generate an Entropy value that can be read via the ENT0-ENT7 registers. The Entropy value may be read once the ENT VAL bit is asserted. Also see ENTa register descriptions (For a = 0 to 7). **NOTE** If PRGM is set to 1, then TRNG_ACC should also be set to 1.
4 DIS_SLF_TST	Disable Self-Tests Writing a 1 to this bit configures various TRNG registers with values that effectively disable internal self-tests. This bit is writable only if PRGM bit is 1, or, PRGM bit is being written to 1 simultaneously while writing to this bit. This bit has lower priority if both this and MCTL[RST_DEF] bits are being written simultaneously. Reading this bit always produces a 0. This is a self-clearing bit.
3-2 OSC_DIV	Oscillator1 Divide Determines the amount of dividing done to the ring oscillator before it is used by the TRNG. This field is writable only if PRGM bit is 1, or PRGM bit is being written to 1 simultaneously while writing this field. This field is reset by writing the MCTL[RST_DEF] bit to 1. Writing 1 to bit MCTL[DIS_SLF_TST] loads a value of 0x0 in this register.

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
	00b - use ring oscillator with no divide 01b - use ring oscillator divided-by-2 10b - use ring oscillator divided-by-4 11b - use ring oscillator divided-by-8
1-0 —	Reserved

41.7.1.3 Statistical Check Miscellaneous Register (SCMISC)

Offset

Register	Offset
SCMISC	4h

Function

The Statistical Check Miscellaneous Register contains the Long Run Maximum Limit value and the Retry Count value. This register is accessible only when the MCTL[PRGM] bit is 1, otherwise this register will read zeros, and cannot be written.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved												RTY_CT			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								LRUN_MAX							
Reset	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1

Fields

Field	Function
31-20 —	Reserved

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
19-16 RTY_CT	Retry count. If a statistical check fails during the TRNG Entropy Generation, the RTY_CT value indicates the number of times a retry should occur before generating an error. This field is writable only if MCTL[PRGM] bit is 1. This field will read zeros if MCTL[PRGM] = 0. This field is reset by writing the MCTL[RST_DEF] bit to 1. Writing 1 to bit MCTL[DIS_SLF_TST] loads a value of 0x1 in this field.
15-8 —	Reserved
7-0 LRUN_MAX	Long run max limit. This value is the largest allowable number of consecutive samples of all 1, or all 0, that is allowed during Entropy generation. This field is writable only if MCTL[PRGM] bit is 1. This field will read zeros if MCTL[PRGM] = 0. This field is reset by writing the MCTL[RST_DEF] bit to 1. Writing 1 to bit MCTL[DIS_SLF_TST] loads a value of 0xFF in this field.

41.7.1.4 Poker Range Register (PKRRNG)

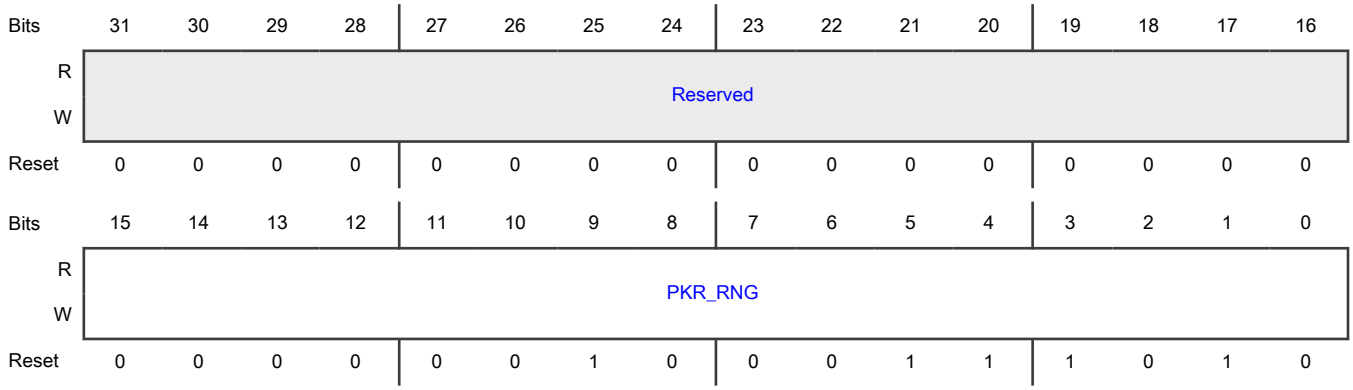
Offset

Register	Offset
PKRRNG	8h

Function

The Poker Range Register defines the difference between the TRNG Poker Maximum Limit and the minimum limit. These limits are used during the TRNG Statistical Check Poker Test. This field is reset by writing the MCTL[RST_DEF] bit to 1. Writing 1 to bit MCTL[DIS_SLF_TST] loads a value of 0xFFFF in this field.

Diagram



Fields

Field	Function
31-16	Reserved

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
—	
15-0 PKR_RNG	Poker Range. During the TRNG Statistical Checks, a "Poker Test" is run which requires a maximum and minimum limit. The maximum is programmed in the PKRMAX[PKR_MAX] register, and the minimum is derived by subtracting the PKR_RNG value from the programmed maximum value. This field is writable only if MCTL[PRGM] bit is 1. This field will read zeros if MCTL[PRGM] = 0. This field is reset by writing the MCTL[RST_DEF] bit to 1. Writing 1 to bit MCTL[DIS_SLF_TST] loads a value of 0xFFFF in this field. Note that the minimum allowable Poker result is PKR_MAX - PKR_RNG + 1.

41.7.1.5 Poker Maximum Limit Register (PKRMAX)

Offset

Register	Offset	Description
PKRMAX	Ch	Accessible at this address when MCTL[PRGM] = 1

Function

The Poker Maximum Limit Register defines Maximum Limit allowable during the TRNG Statistical Check Poker Test. Note that this offset (0x0C) is used as PKRMAX only if MCTL[PRGM] is 1. If MCTL[PRGM] is 0, this offset is used as the PKRSQ read-back register.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0								PKR_MAX							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PKR_MAX															
W																
Reset	0	0	0	0	0	1	1	0	0	1	0	0	0	0	0	0

Fields

Field	Function
31-24 —	Reserved
23-0	Poker Maximum Limit.

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
PKR_MAX	During the TRNG Statistical Checks, a "Poker Test" is run which requires a maximum and minimum limit. The maximum allowable result is programmed in the PKRMAX[PKR_MAX] register. This field is writable only if MCTL[PRGM] bit is 1. This register is reset by writing the MCTL[RST_DEF] bit to 1. Writing 1 to bit MCTL[DIS_SLF_TST] loads a value of 0xFFFF in this field. Note that the PKRMAX and PKRRNG registers combined are used to define the minimum allowable Poker result, which is PKR_MAX - PKR_RNG + 1. Note that if MCTL[PRGM] bit is 0, this register address is used to read the Poker Test Square Calculation result in register PKRSQ, as defined in the following section.

41.7.1.6 Poker Square Calculation Result Register (PKRSQ)

Offset

Register	Offset	Description
PKRSQ	Ch	Accessible at this address when MCTL[PRGM] = 0

Function

The Poker Square Calculation Result Register is a read-only register used to read the result of the TRNG Statistical Check Poker Test's Square Calculation. This test starts with the PKRMAX value and decreases toward a final result, which is read here. For the Poker Test to pass, this final result must be less than the programmed PKRRNG value. Note that this offset (0x0C) is used as PKRMAX if MCTL[PRGM] is 1. If MCTL[PRGM] is 0, this offset is used as PKRSQ read back register, as described here.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0								PKR_SQ							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PKR_SQ															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-24 —	Reserved
23-0	Poker Square Calculation Result.

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
PKR_SQ	During the TRNG Statistical Checks, a "Poker Test" is run which starts with the value PKRMAX[PKR_MAX]. This value decreases according to a "sum of squares" algorithm, and must remain greater than zero, but less than the PKRRNG[PKR_RNG] limit. The resulting value may be read through this register, if MCTL[PRGM] bit is 0. Note that if MCTL[PRGM] bit is 1, this register address is used to access the Poker Test Maximum Limit in register PKRMAX, as defined in the previous section.

41.7.1.7 Seed Control Register (SDCTL)

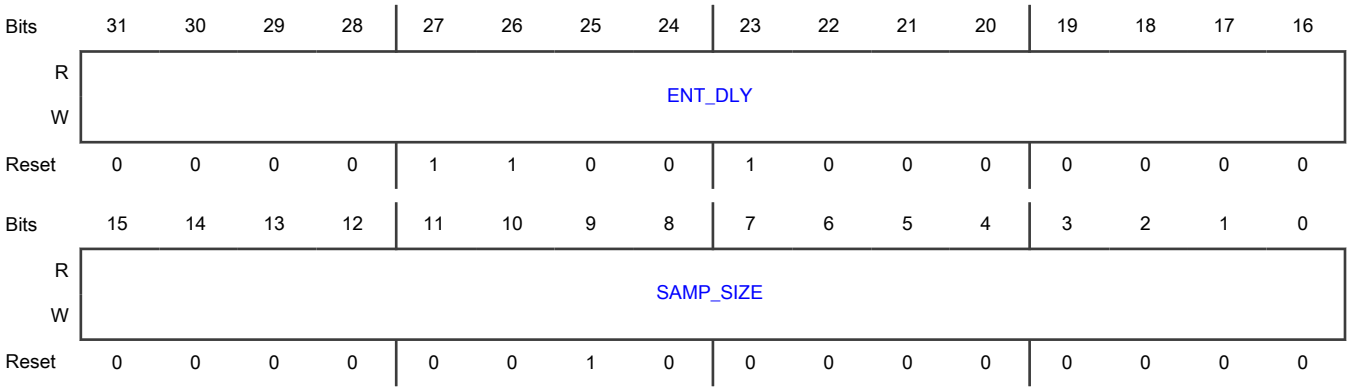
Offset

Register	Offset
SDCTL	10h

Function

The Seed Control Register contains two fields. One field defines the length (in system clocks) of each Entropy sample (ENT_DLY), and the other field indicates the number of samples that will be taken during each TRNG Entropy generation (SAMP_SIZE).

Diagram



Fields

Field	Function
31-16 ENT_DLY	Entropy Delay. Defines the length (in system clocks) of each Entropy sample taken. This field is writable only if MCTL[PRGM] bit is 1. This field will read zeros if MCTL[PRGM] = 0. This field is reset upon assertion of TRNG's asynchronous reset input indicating a functional reset or Power-On Reset (POR).
15-0 SAMP_SIZE	Sample Size. Defines the total number of Entropy samples that will be taken during Entropy generation. This field is writable only if MCTL[PRGM] bit is 1. This field will read zeros if MCTL[PRGM] = 0. This field is reset by writing the MCTL[RST_DEF] bit to 1.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

41.7.1.8 Sparse Bit Limit Register (SBLIM)

Offset

Register	Offset	Description
SBLIM	14h	Accessible at this address when MCTL[PRGM] = 1

Function

The Sparse Bit Limit Register is used when Von Neumann sampling is selected during Entropy Generation. It defines the maximum number of consecutive Von Neumann samples which may be discarded before an error is generated. Note that this address (0x14) is used as SBLIM only if MCTL[PRGM] is 1. If MCTL[PRGM] is 0, this address is used as TOTSAM read back register.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved						SB_LIM									
Reset	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1

Fields

Field	Function
31-10 —	Reserved
9-0 SB_LIM	Sparse Bit Limit. During Von Neumann sampling (if enabled by MCTL[SAMP_MODE], samples are discarded if two consecutive raw samples are both 0 or both 1. If this discarding occurs for a long period of time, it indicates that there is insufficient Entropy. The Sparse Bit Limit defines the maximum number of consecutive samples that may be discarded before an error is generated. This field is writable only if MCTL[PRGM] bit is 1. This register is reset by writing the MCTL[RST_DEF] bit to 1. Writing 1 to bit MCTL[DIS_SLF_TST] loads a value of 0x3F in this field. Note that if MCTL[PRGM] bit is 0, this register address is used to read the Total Samples count in register TOTSAM, as defined in the following section.

41.7.1.9 Total Samples Register (TOTSAM)

Offset

Register	Offset	Description
TOTSAM	14h	Accessible at this address when MCTL[PRGM] = 0

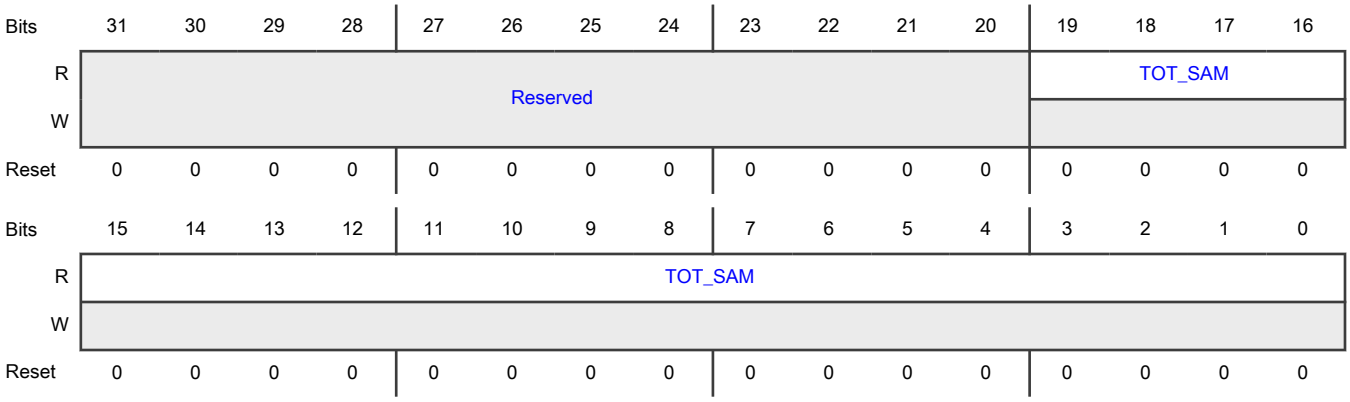
All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Function

The Total Samples Register is a read-only register used to read the total number of samples taken during Entropy generation. It is used to give an indication of how often a sample is actually used during Von Neumann sampling. Note that this offset (0x14) is used as SBLIM if MCTL[PRGM] is 1. If MCTL[PRGM] is 0, this offset is used as TOTSAM read back register, as described here.

Diagram



Fields

Field	Function
31-20 —	Reserved
19-0 TOT_SAM	Total Samples. During Entropy generation, the total number of raw samples is counted. This count is useful in determining how often a sample is used during Von Neumann sampling. The count may be read through this register, if MCTL[PRGM] bit is 0.

41.7.1.10 Frequency Count Minimum Limit Register (FRQMIN)

Offset

Register	Offset	Description
FRQMIN	18h	Accessible at this address when MCTL[PRGM] = 1

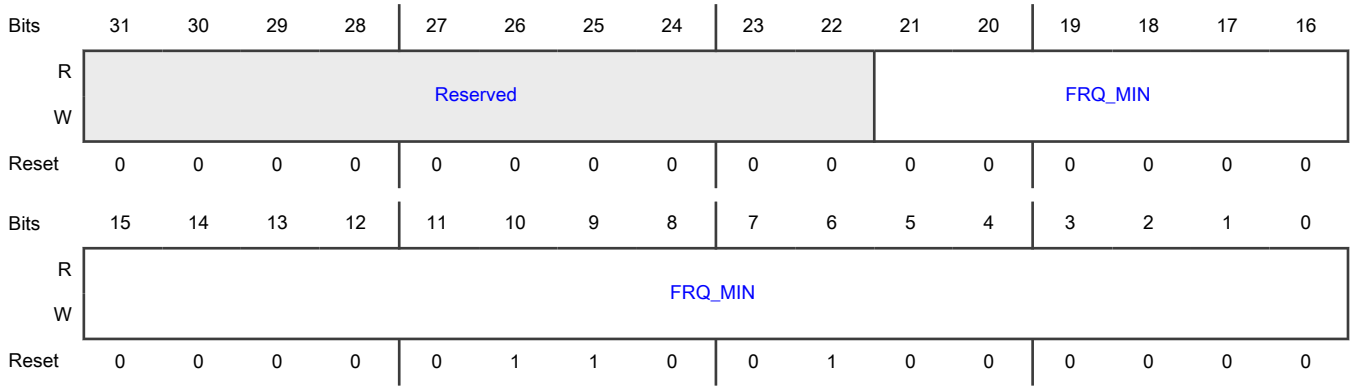
Function

The Frequency Count Minimum Limit Register defines the minimum allowable count taken by the Entropy sample counter during each Entropy sample. During any sample period, if the count is less than this programmed minimum, a Frequency Count Fail is flagged in MCTL[FCT_FAIL] and an error is generated. Note that this address (0x18) is used as FRQMIN only if MCTL[PRGM] is 1. If MCTL[PRGM] is 0, this address is used as OSC2_FRQCNT read back register when OSC2_CTL[TRNG_ENT_CTL] = 10b. Writing 1 to bit MCTL[DIS_SLF_TST] loads a value of 0x0 in this field.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram



Fields

Field	Function
31-22 —	Reserved
21-0 FRQ_MIN	Frequency Count Minimum Limit. Defines the minimum allowable count taken during each entropy sample. This field is writable only if MCTL[PRGM] bit is 1. This field will read zeros if MCTL[PRGM] = 0. This field is cleared to its reset value at POR. Note that if MCTL[PRGM] bit is 0 and OSC2_CTL[TRNG_ENT_CTL] = 10b, this register address is used to read the Oscillator-2 Frequency Count result in register OSC2_FRQCNT, as defined in the OSC2_FRQ_CT field. Writing 1 to bit MCTL[DIS_SLF_TST] loads a value of 0x0 in this field.

41.7.1.11 Oscillator-2 Frequency Count Register (OSC2_FRQCNT)

Offset

Register	Offset	Description
OSC2_FRQCNT	18h	Accessible at this address when MCTL[PRGM] = 0

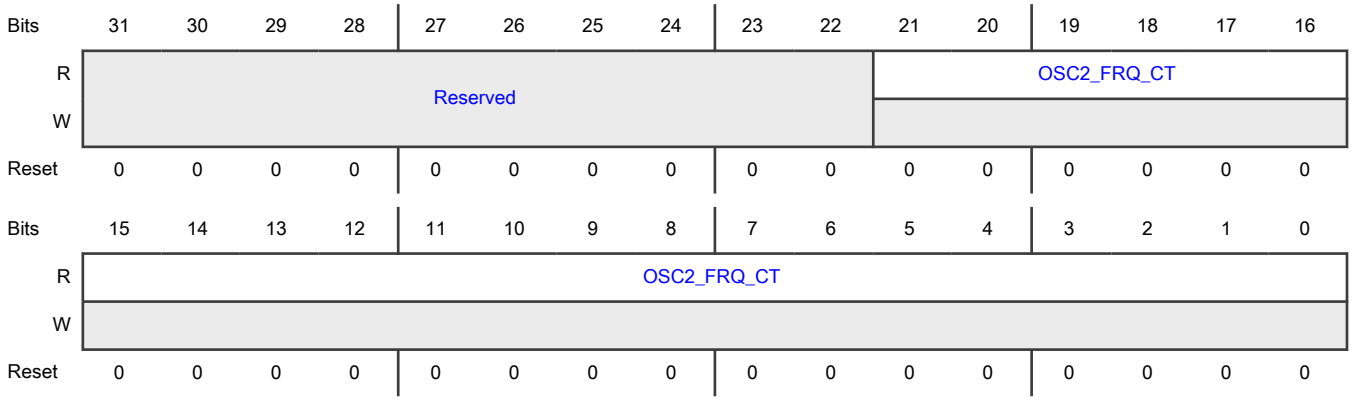
Function

The Oscillator-2 Frequency Count Register is a read-only register used to read the oscillator-2 frequency counter within the TRNG entropy generator. It will read all zeros unless MCTL[PRGM] = 0 and MCTL[TRNG_ACC] = 1. Note that this offset (0x18) is used as FRQMIN if MCTL[PRGM] is 1. If MCTL[PRGM] is 0 and OSC2_CTL[TRNG_ENT_CTL] = 10b, this offset is used as live OSC2_FRQCNT read back register, as described here.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram



Fields

Field	Function
31-22 —	Reserved
21-0 OSC2_FRQ_CT	Frequency Count. If MCTL[TRNG_ACC] = 1, Reads a sample frequency count taken during entropy generation when OSC2_CTL[TRNG_ENT_CTL] = 10b. Requires MCTL[PRGM] = 0. Note that if MCTL[PRGM] bit is 1, this register address is used to access the Frequency Minimum Limit in register FRQMIN, as defined in the FRQ_MIN field. This register must be read once, as a full 32-bit value. A subsequent read is only valid when MCTL[FCT_VAL] bit is set. A read clears this register. The next read when MCTL[FCT_VAL] bit is set must be treated as a different value from the previous one, even though the counts might be the same. If OSC2_CTL[TRNG_ENT_CTL] = 10b, OSC2_FRQ_CT shows the frequency count from TRNG oscillator 2. If OSC2_CTL[TRNG_ENT_CTL] = 00b or 01b, OSC2_FRQ_CT is 0.

41.7.1.12 Frequency Count Register (FRQCNT)

Offset

Register	Offset	Description
FRQCNT	1Ch	Accessible at this address when MCTL[PRGM] = 0

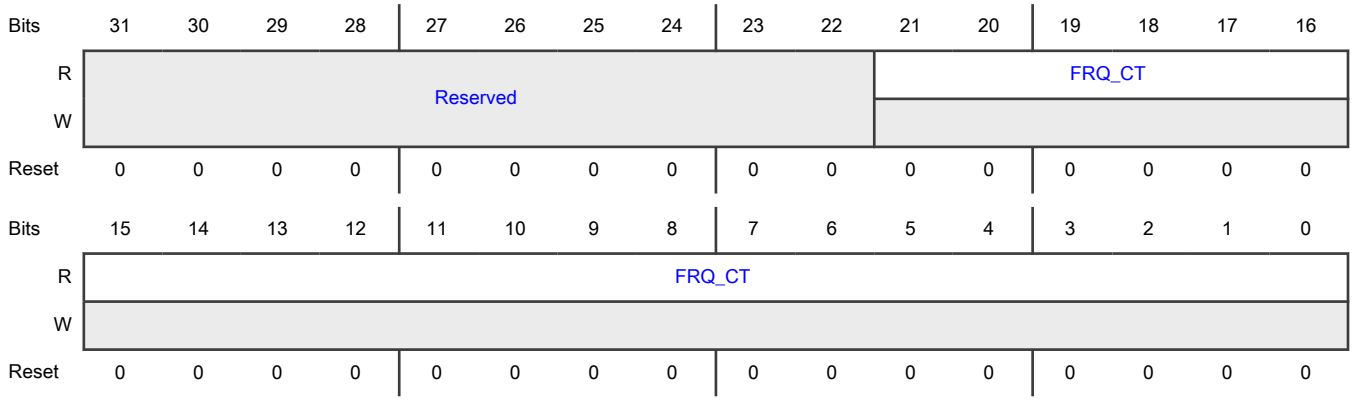
Function

The Frequency Count Register is a read-only register used to read the frequency counter within the TRNG entropy generator. It will read all zeros unless MCTL[PRGM] = 0 and MCTL[TRNG_ACC] = 1. Note that this offset (0x1C) is used as FRQMAX if MCTL[PRGM] is 1. If MCTL[PRGM] is 0, this offset is used as live FRQCNT read back register, as described here.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram



Fields

Field	Function
31-22 —	Reserved
21-0 FRQ_CT	Frequency Count. If MCTL[TRNG_ACC] = 1, Reads a sample frequency count taken during entropy generation. Requires MCTL[PRGM] = 0. Note that if MCTL[PRGM] bit is 1, this register address is used to access the Frequency Maximum Limit in register FRQMAX, as defined in the FRQ_MAX field. This register must be read once, as a full 32-bit value. A subsequent read is only valid when MCTL[FCT_VAL] bit is set. A read clears this register. The next read when MCTL[FCT_VAL] bit is set must be treated as a different value from the previous one, even though the counts might be the same. If OSC2_CTL[TRNG_ENT_CTL] = 00b or 01b, FRQ_CT shows the frequency count from TRNG oscillator 1. If OSC2_CTL[TRNG_ENT_CTL] = 10b, FRQ_CT = 0, unless MCTL[FCT_FAIL] bit is set in which case it shows the frequency count from TRNG oscillator 2.

41.7.1.13 Frequency Count Maximum Limit Register (FRQMAX)

Offset

Register	Offset	Description
FRQMAX	1Ch	Accessible at this address when MCTL[PRGM] = 1

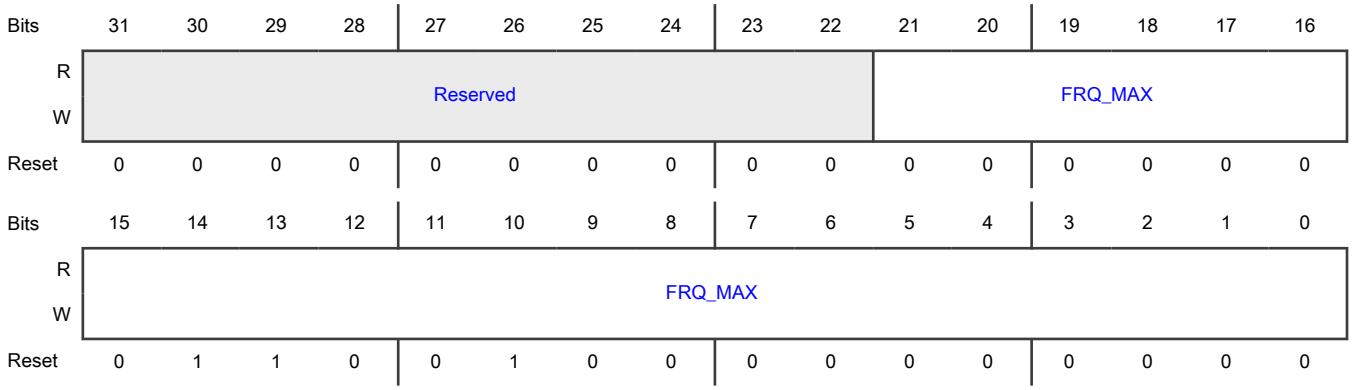
Function

The Frequency Count Maximum Limit Register defines the maximum allowable count taken by the Entropy sample counter during each Entropy sample. During any sample period, if the count is greater than this programmed maximum, a Frequency Count Fail is flagged in MCTL[FCT_FAIL] and an error is generated. Note that this address (0x1C) is used as FRQMAX only if MCTL[PRGM] is 1. If MCTL[PRGM] is 0, this address is used as FRQCNT read back register. Writing 1 to bit MCTL[DIS_SLF_TST] loads a value of 0x3FFFFF in this field.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram



Fields

Field	Function
31-22 —	Reserved
21-0 FRQ_MAX	Frequency Counter Maximum Limit. Defines the maximum allowable count taken during each entropy sample. This field is writable only if MCTL[PRGM] bit is 1. This field is cleared to its reset value at POR. Note that if MCTL[PRGM] bit is 0, this register address is used to read the Frequency Count result in register FRQCNT, as defined in the FRQ_CT field. Writing 1 to bit MCTL[DIS_SLF_TST] loads a value of 0x3FFFFF in this field.

41.7.1.14 Statistical Check Monobit Count Register (SCMC)

Offset

Register	Offset	Description
SCMC	20h	Accessible at this address when MCTL[PRGM] = 0

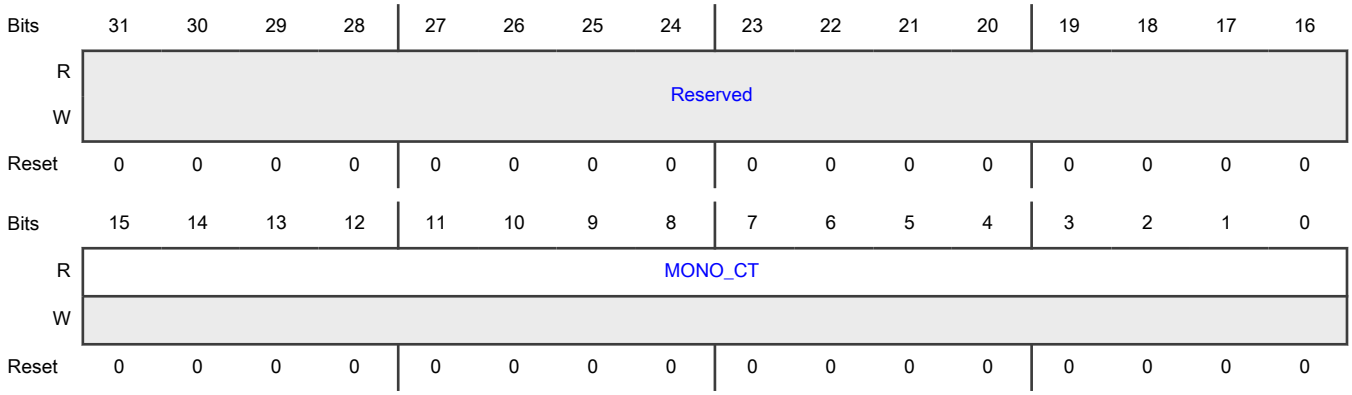
Function

The Statistical Check Monobit Count Register is a read-only register used to read the final monobit count after entropy generation. This counter starts with the value in SCML[MONO_MAX], and is decremented each time a one is sampled. Note that this offset (0x20) is used as SCML if MCTL[PRGM] is 1. If MCTL[PRGM] is 0, this offset is used as SCMC read back register, as described here.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram



Fields

Field	Function
31-16 —	Reserved
15-0 MONO_CT	Monobit Count. Reads the final Monobit count after entropy generation. Requires MCTL[PRGM] = 0. Note that if MCTL[PRGM] bit is 1, this register address is used to access the Statistical Check Monobit Limit in register SCML, as defined in the next section.

41.7.1.15 Statistical Check Monobit Limit Register (SCML)

Offset

Register	Offset	Description
SCML	20h	Accessible at this address when MCTL[PRGM] = 1

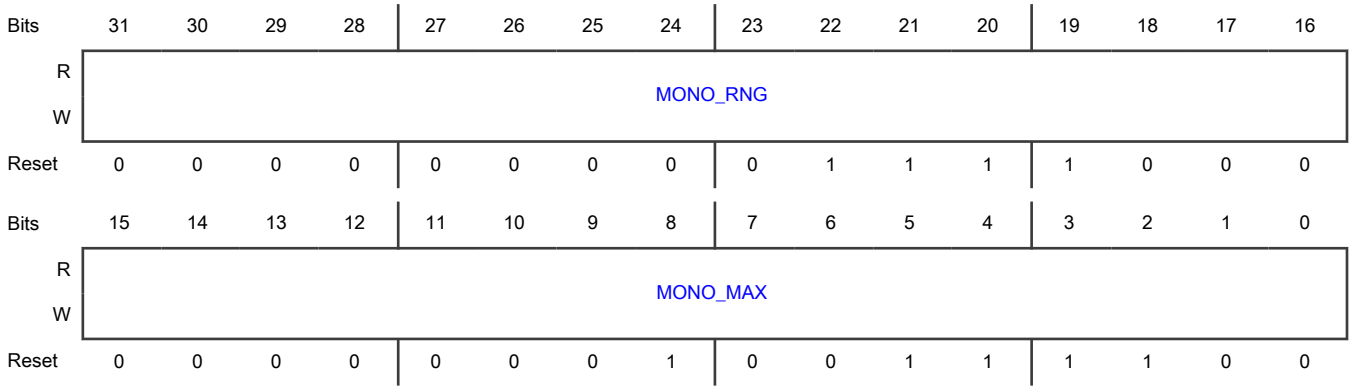
Function

The Statistical Check Monobit Limit Register defines the allowable maximum and minimum number of ones/zeros detected during entropy generation. To pass the test, the number of ones/zeros generated must be less than the programmed maximum value, and the number of ones/zeros generated must be greater than (MONO_MAX - MONO_RNG). If this test fails, the Retry Counter in SCMISC will be decremented, and a retry will occur if the Retry Count has not reached zero. If the Retry Count has reached zero, an error will be generated. Note that this offset (0x20) is used as SCML only if MCTL[PRGM] is 1. If MCTL[PRGM] is 0, this offset is used as SCMC read back register.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram



Fields

Field	Function
31-16 MONO_RNG	Monobit Range. The number of ones/zeros detected during entropy generation must be greater than MONO_MAX - MONO_RNG, else a retry or error will occur. This register is cleared to the reset value by writing the MCTL[RST_DEF] bit to 1. Writing 1 to bit MCTL[DIS_SLF_TST] loads a value of 0xFFFF in this field.
15-0 MONO_MAX	Monobit Maximum Limit. Defines the maximum allowable count taken during entropy generation. The number of ones/zeros detected during entropy generation must be less than MONO_MAX, else a retry or error will occur. This register is reset by writing the MCTL[RST_DEF] bit to 1. Writing 1 to bit MCTL[DIS_SLF_TST] loads a value of 0xFFFF in this field.

41.7.1.16 Statistical Check Run Length 1 Count Register (SCR1C)

Offset

Register	Offset	Description
SCR1C	24h	Accessible at this address when MCTL[PRGM] = 0

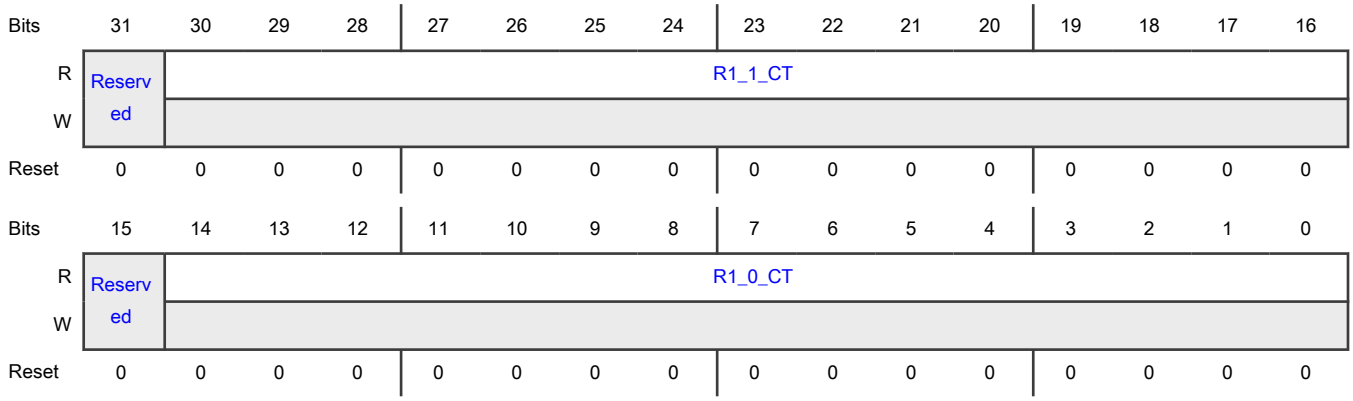
Function

The Statistical Check Run Length 1 Counters Register is a read-only register used to read the final Run Length 1 counts after entropy generation. These counters start with the value in SCR1L[RUN1_MAX]. The R1_1_CT decrements each time a single one is sampled (preceded by a zero and followed by a zero). The R1_0_CT decrements each time a single zero is sampled (preceded by a one and followed by a one). Note that this offset (0x24) is used as SCR1L if MCTL[PRGM] is 1. If MCTL[PRGM] is 0, this offset is used as SCR1C read back register, as described here.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram



Fields

Field	Function
31 —	Reserved
30-16 R1_1_CT	Runs of One, Length 1 Count Reads the final Runs of Ones, length 1 count after entropy generation. Requires MCTL[PRGM] = 0.
15 —	Reserved
14-0 R1_0_CT	Runs of Zero, Length 1 Count Reads the final Runs of zeros, length 1 count after entropy generation. Requires MCTL[PRGM] = 0.

41.7.1.17 Statistical Check Run Length 1 Limit Register (SCR1L)

Offset

Register	Offset	Description
SCR1L	24h	Accessible at this address when MCTL[PRGM] = 1

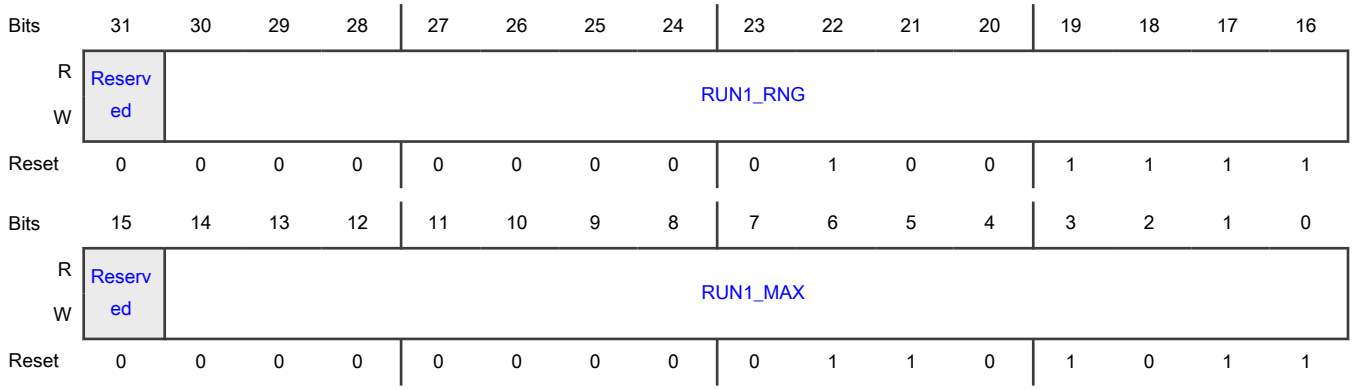
Function

The Statistical Check Run Length 1 Limit Register defines the allowable maximum and minimum number of runs of length 1 detected during entropy generation. To pass the test, the number of runs of length 1 (for samples of both 0 and 1) must be less than the programmed maximum value, and the number of runs of length 1 must be greater than (RUN1_MAX - RUN1_RNG). If this test fails, the Retry Counter in SCMISC will be decremented, and a retry will occur if the Retry Count has not reached zero. If the Retry Count has reached zero, an error will be generated. Note that this address (0x24) is used as SCR1L only if MCTL[PRGM] is 1. If MCTL[PRGM] is 0, this address is used as SCR1C read back register.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram



Fields

Field	Function
31 —	Reserved
30-16 RUN1_RNG	Run Length 1 Range The number of runs of length 1 (for both 0 and 1) detected during entropy generation must be greater than RUN1_MAX - RUN1_RNG, else a retry or error will occur. This register is reset by writing the MCTL[RST_DEF] bit to 1. Writing 1 to bit MCTL[DIS_SLF_TST] loads a value of 0x7FFF in this field.
15 —	Reserved
14-0 RUN1_MAX	Run Length 1 Maximum Limit Defines the maximum allowable runs of length 1 (for both 0 and 1) detected during entropy generation. The number of runs of length 1 detected during entropy generation must be less than RUN1_MAX, else a retry or error will occur. This register is reset by writing the MCTL[RST_DEF] bit to 1. Writing 1 to bit MCTL[DIS_SLF_TST] loads a value of 0x7FFE in this field.

41.7.1.18 Statistical Check Run Length 2 Count Register (SCR2C)

Offset

Register	Offset	Description
SCR2C	28h	Accessible at this address when MCTL[PRGM] = 0

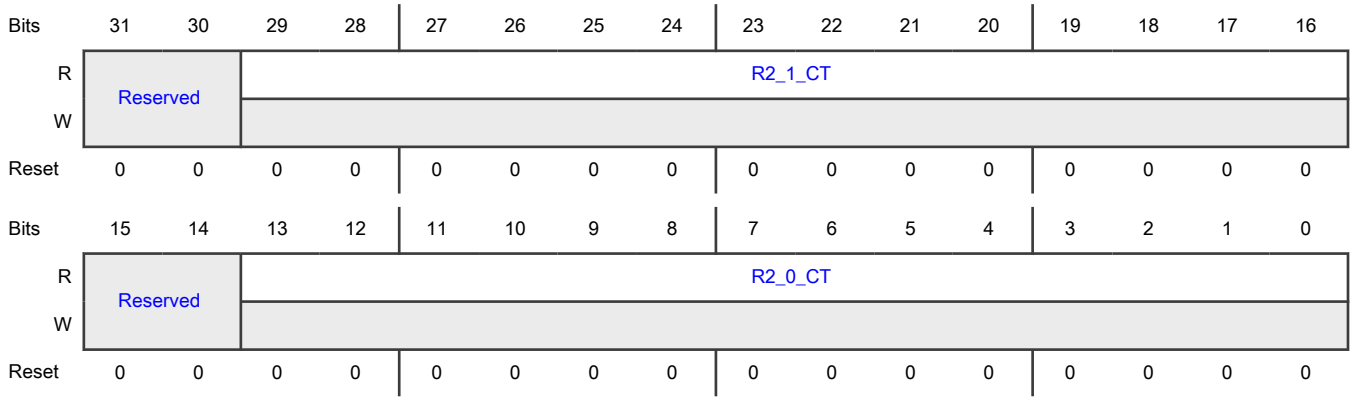
Function

The Statistical Check Run Length 2 Counters Register is a read-only register used to read the final Run Length 2 counts after entropy generation. These counters start with the value in SCR2L[RUN2_MAX]. The R2_1_CT decrements each time two consecutive ones are sampled (preceded by a zero and followed by a zero). The R2_0_CT decrements each time two consecutive zeros are sampled (preceded by a one and followed by a one). Note that this offset (0x28) is used as SCR2L if MCTL[PRGM] is 1. If MCTL[PRGM] is 0, this offset is used as SCR2C read back register, as described here.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram



Fields

Field	Function
31-30 —	Reserved
29-16 R2_1_CT	Runs of One, Length 2 Count. Reads the final Runs of Ones, length 2 count after entropy generation. Requires MCTL[PRGM] = 0.
15-14 —	Reserved
13-0 R2_0_CT	Runs of Zero, Length 2 Count. Reads the final Runs of zeros, length 2 count after entropy generation. Requires MCTL[PRGM] = 0.

41.7.1.19 Statistical Check Run Length 2 Limit Register (SCR2L)

Offset

Register	Offset	Description
SCR2L	28h	Accessible at this address when MCTL[PRGM] = 1

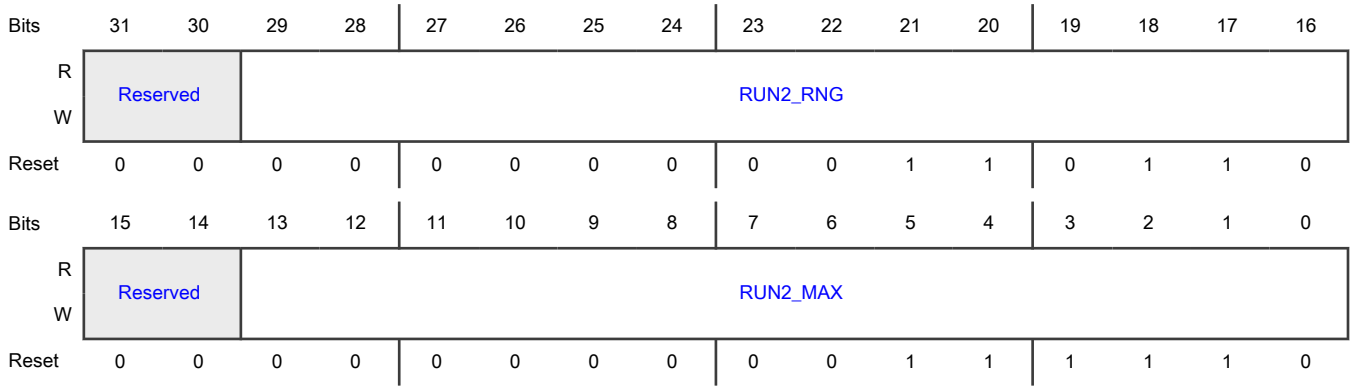
Function

The Statistical Check Run Length 2 Limit Register defines the allowable maximum and minimum number of runs of length 2 detected during entropy generation. To pass the test, the number of runs of length 2 (for samples of both 0 and 1) must be less than the programmed maximum value, and the number of runs of length 2 must be greater than (RUN2_MAX - RUN2_RNG). If this test fails, the Retry Counter in SCMISC will be decremented, and a retry will occur if the Retry Count has not reached zero. If the Retry Count has reached zero, an error will be generated. Note that this address (0x28) is used as SCR2L only if MCTL[PRGM] is 1. If MCTL[PRGM] is 0, this address is used as SCR2C read back register.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram



Fields

Field	Function
31-30 —	Reserved
29-16 RUN2_RNG	Run Length 2 Range. The number of runs of length 2 (for both 0 and 1) detected during entropy generation must be greater than RUN2_MAX - RUN2_RNG, else a retry or error will occur. This register is reset by writing the MCTL[RST_DEF] bit to 1. Writing 1 to bit MCTL[DIS_SLF_TST] loads a value of 0x3FFF in this field.
15-14 —	Reserved
13-0 RUN2_MAX	Run Length 2 Maximum Limit. Defines the maximum allowable runs of length 2 (for both 0 and 1) detected during entropy generation. The number of runs of length 2 detected during entropy generation must be less than RUN2_MAX, else a retry or error will occur. This register is reset by writing the MCTL[RST_DEF] bit to 1. Writing 1 to bit MCTL[DIS_SLF_TST] loads a value of 0x3FFE in this field.

41.7.1.20 Statistical Check Run Length 3 Count Register (SCR3C)

Offset

Register	Offset	Description
SCR3C	2Ch	Accessible at this address when MCTL[PRGM] = 0

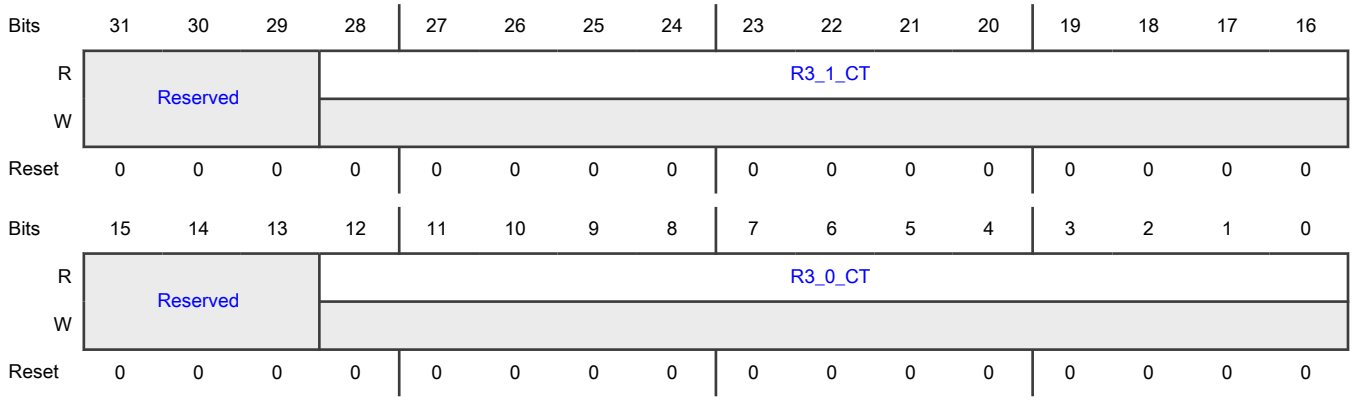
Function

The Statistical Check Run Length 3 Counters Register is a read-only register used to read the final Run Length 3 counts after entropy generation. These counters start with the value in SCR3L[RUN3_MAX]. The R3_1_CT decrements each time three consecutive ones are sampled (preceded by a zero and followed by a zero). The R3_0_CT decrements each time three consecutive zeros are sampled (preceded by a one and followed by a one). Note that this offset (0x2C) is used as SCR3L if MCTL[PRGM] is 1. If MCTL[PRGM] is 0, this offset is used as SCR3C read back register, as described here.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram



Fields

Field	Function
31-29 —	Reserved
28-16 R3_1_CT	Runs of Ones, Length 3 Count. Reads the final Runs of Ones, length 3 count after entropy generation. Requires MCTL[PRGM] = 0.
15-13 —	Reserved
12-0 R3_0_CT	Runs of zeros, Length 3 Count. Reads the final Runs of zeros, length 3 count after entropy generation. Requires MCTL[PRGM] = 0.

41.7.1.21 Statistical Check Run Length 3 Limit Register (SCR3L)

Offset

Register	Offset	Description
SCR3L	2Ch	Accessible at this address when MCTL[PRGM] = 1

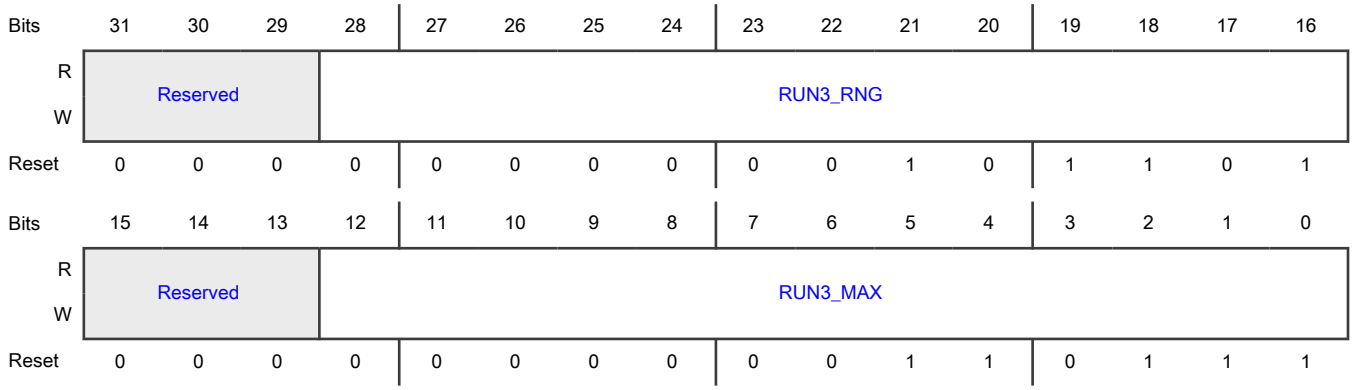
Function

The Statistical Check Run Length 3 Limit Register defines the allowable maximum and minimum number of runs of length 3 detected during entropy generation. To pass the test, the number of runs of length 3 (for samples of both 0 and 1) must be less than the programmed maximum value, and the number of runs of length 3 must be greater than (RUN3_MAX - RUN3_RNG). If this test fails, the Retry Counter in SCMISC will be decremented, and a retry will occur if the Retry Count has not reached zero. If the Retry Count has reached zero, an error will be generated. Note that this address (0x2C) is used as SCR3L only if MCTL[PRGM] is 1. If MCTL[PRGM] is 0, this address is used as SCR3C read back register.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram



Fields

Field	Function
31-29 —	Reserved
28-16 RUN3_RNG	Run Length 3 Range. The number of runs of length 3 (for both 0 and 1) detected during entropy generation must be greater than RUN3_MAX - RUN3_RNG, else a retry or error will occur. This register is reset by writing the MCTL[RST_DEF] bit to 1. Writing 1 to bit MCTL[DIS_SLF_TST] loads a value of 0x1FFF in this field.
15-13 —	Reserved
12-0 RUN3_MAX	Run Length 3 Maximum Limit. Defines the maximum allowable runs of length 3 (for both 0 and 1) detected during entropy generation. The number of runs of length 3 detected during entropy generation must be less than RUN3_MAX, else a retry or error will occur. This register is reset by writing the MCTL[RST_DEF] bit to 1. Writing 1 to bit MCTL[DIS_SLF_TST] loads a value of 0x1FFE in this field.

41.7.1.22 Statistical Check Run Length 4 Count Register (SCR4C)

Offset

Register	Offset	Description
SCR4C	30h	Accessible at this address when MCTL[PRGM] = 0

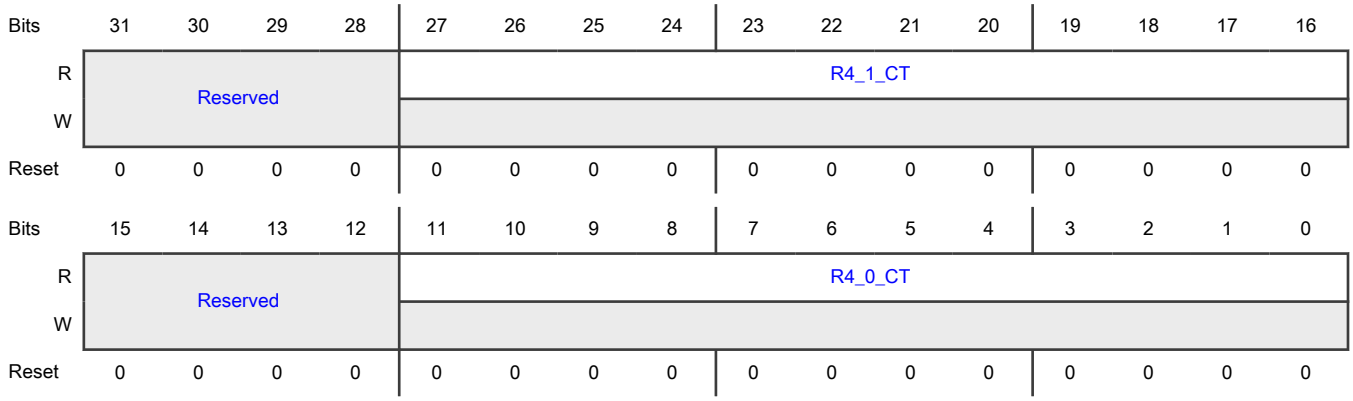
Function

The Statistical Check Run Length 4 Counters Register is a read-only register used to read the final Run Length 4 counts after entropy generation. These counters start with the value in SCR4L[RUN4_MAX]. The R4_1_CT decrements each time four consecutive ones are sampled (preceded by a zero and followed by a zero). The R4_0_CT decrements each time four consecutive zeros are sampled (preceded by a one and followed by a one). Note that this offset (0x30) is used as SCR4L if MCTL[PRGM] is 1. If MCTL[PRGM] is 0, this offset is used as SCR4C read back register, as described here.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram



Fields

Field	Function
31-28 —	Reserved
27-16 R4_1_CT	Runs of One, Length 4 Count. Reads the final Runs of Ones, length 4 count after entropy generation. Requires MCTL[PRGM] = 0.
15-12 —	Reserved
11-0 R4_0_CT	Runs of Zero, Length 4 Count. Reads the final Runs of Ones, length 4 count after entropy generation. Requires MCTL[PRGM] = 0.

41.7.1.23 Statistical Check Run Length 4 Limit Register (SCR4L)

Offset

Register	Offset	Description
SCR4L	30h	Accessible at this address when MCTL[PRGM] = 1

Function

The Statistical Check Run Length 4 Limit Register defines the allowable maximum and minimum number of runs of length 4 detected during entropy generation. To pass the test, the number of runs of length 4 (for samples of both 0 and 1) must be less than the programmed maximum value, and the number of runs of length 4 must be greater than (RUN4_MAX - RUN4_RNG). If this test fails, the Retry Counter in SCMISC will be decremented, and a retry will occur if the Retry Count has not reached zero. If the Retry Count has reached zero, an error will be generated. Note that this address (0x30) is used as SCR4L only if MCTL[PRGM] is 1. If MCTL[PRGM] is 0, this address is used as SCR4C read back register.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved				RUN4_RNG											
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved				RUN4_MAX											
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	0

Fields

Field	Function
31-28 —	Reserved
27-16 RUN4_RNG	Run Length 4 Range. The number of runs of length 4 (for both 0 and 1) detected during entropy generation must be greater than RUN4_MAX - RUN4_RNG, else a retry or error will occur. This register is reset by writing the MCTL[RST_DEF] bit to 1. Writing 1 to bit MCTL[DIS_SLF_TST] loads a value of 0xFFF in this field.
15-12 —	Reserved
11-0 RUN4_MAX	Run Length 4 Maximum Limit. Defines the maximum allowable runs of length 4 (for both 0 and 1) detected during entropy generation. The number of runs of length 4 detected during entropy generation must be less than RUN4_MAX, else a retry or error will occur. This register is reset by writing the MCTL[RST_DEF] bit to 1. Writing 1 to bit MCTL[DIS_SLF_TST] loads a value of 0xFFE in this field.

41.7.1.24 Statistical Check Run Length 5 Count Register (SCR5C)

Offset

Register	Offset	Description
SCR5C	34h	Accessible at this address when MCTL[PRGM] = 0

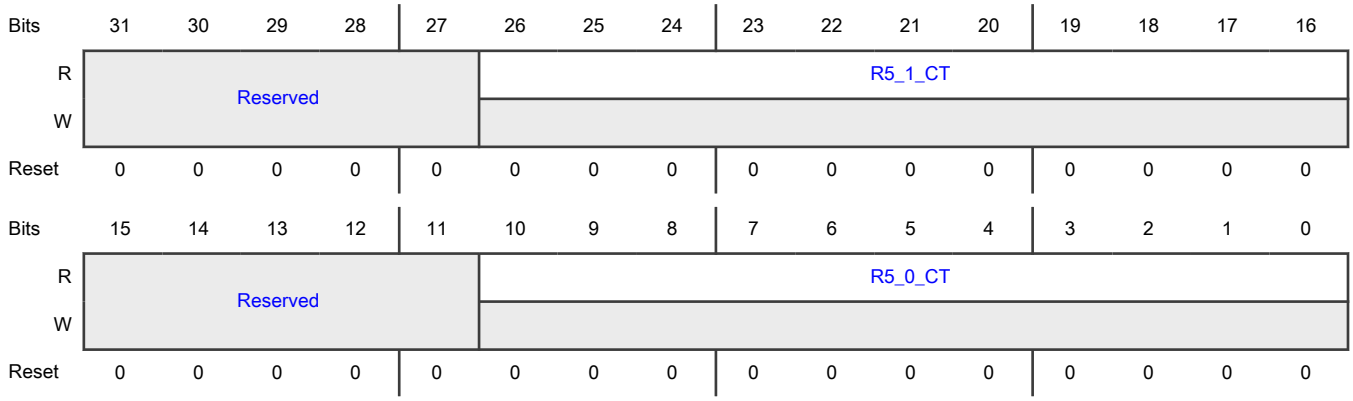
Function

The Statistical Check Run Length 5 Counters Register is a read-only register used to read the final Run Length 5 counts after entropy generation. These counters start with the value in SCR5L[RUN5_MAX]. The R5_1_CT decrements each time five consecutive ones are sampled (preceded by a zero and followed by a zero). The R5_0_CT decrements each time five consecutive zeros are sampled (preceded by a one and followed by a one). Note that this offset (0x34) is used as SCR5L if MCTL[PRGM] is 1. If MCTL[PRGM] is 0, this offset is used as SCR5C read back register, as described here.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram



Fields

Field	Function
31-27 —	Reserved
26-16 R5_1_CT	Runs of One, Length 5 Count. Reads the final Runs of Ones, length 5 count after entropy generation. Requires MCTL[PRGM] = 0.
15-11 —	Reserved
10-0 R5_0_CT	Runs of Zero, Length 5 Count. Reads the final Runs of Ones, length 5 count after entropy generation. Requires MCTL[PRGM] = 0.

41.7.1.25 Statistical Check Run Length 5 Limit Register (SCR5L)

Offset

Register	Offset	Description
SCR5L	34h	Accessible at this address when MCTL[PRGM] = 1

Function

The Statistical Check Run Length 5 Limit Register defines the allowable maximum and minimum number of runs of length 5 detected during entropy generation. To pass the test, the number of runs of length 5 (for samples of both 0 and 1) must be less than the programmed maximum value, and the number of runs of length 5 must be greater than (RUN5_MAX - RUN5_RNG). If this test fails, the Retry Counter in SCMISC will be decremented, and a retry will occur if the Retry Count has not reached zero. If the Retry Count has reached zero, an error will be generated. Note that this address (0x34) is used as SCR5L only if MCTL[PRGM] is 1. If MCTL[PRGM] is 0, this address is used as SCR5C read back register.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved					RUN5_RNG										
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved					RUN5_MAX										
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0

Fields

Field	Function
31-27 —	Reserved
26-16 RUN5_RNG	Run Length 5 Range. The number of runs of length 5 (for both 0 and 1) detected during entropy generation must be greater than RUN5_MAX - RUN5_RNG, else a retry or error will occur. This register is reset by writing the MCTL[RST_DEF] bit to 1. Writing 1 to bit MCTL[DIS_SLF_TST] loads a value of 0x7FF in this field.
15-11 —	Reserved
10-0 RUN5_MAX	Run Length 5 Maximum Limit. Defines the maximum allowable runs of length 5 (for both 0 and 1) detected during entropy generation. The number of runs of length 5 detected during entropy generation must be less than RUN5_MAX, else a retry or error will occur. This register is reset by writing the MCTL[RST_DEF] bit to 1. Writing 1 to bit MCTL[DIS_SLF_TST] loads a value of 0x7FE in this field.

41.7.1.26 Statistical Check Run Length 6+ Count Register (SCR6PC)

Offset

Register	Offset	Description
SCR6PC	38h	Accessible at this address when MCTL[PRGM] = 0

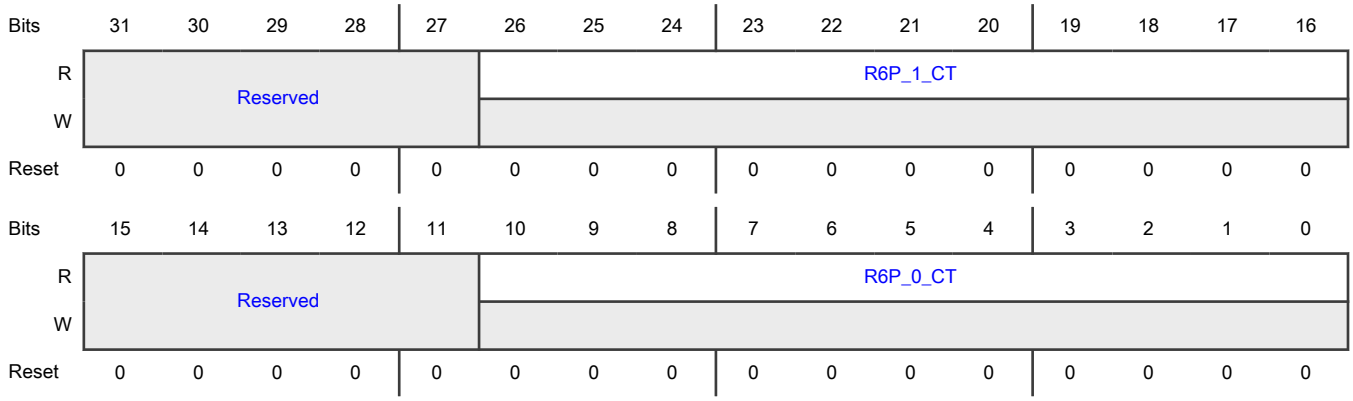
Function

The Statistical Check Run Length 6+ Counters Register is a read-only register used to read the final Run Length 6+ counts after entropy generation. These counters start with the value in SCR6PL[RUN6P_MAX]. The R6P_1_CT decrements each time six or more consecutive ones are sampled (preceded by a zero and followed by a zero). The R6P_0_CT decrements each time six or more consecutive zeros are sampled (preceded by a one and followed by a one). Note that this offset (0x38) is used as SCR6PL if MCTL[PRGM] is 1. If MCTL[PRGM] is 0, this offset is used as SCR6PC read back register, as described here.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram



Fields

Field	Function
31-27 —	Reserved
26-16 R6P_1_CT	Runs of One, Length 6+ Count. Reads the final Runs of Ones, length 6+ count after entropy generation. Requires MCTL[PRGM] = 0.
15-11 —	Reserved
10-0 R6P_0_CT	Runs of Zero, Length 6+ Count. Reads the final Runs of Ones, length 6+ count after entropy generation. Requires MCTL[PRGM] = 0.

41.7.1.27 Statistical Check Run Length 6+ Limit Register (SCR6PL)

Offset

Register	Offset	Description
SCR6PL	38h	Accessible at this address when MCTL[PRGM] = 1

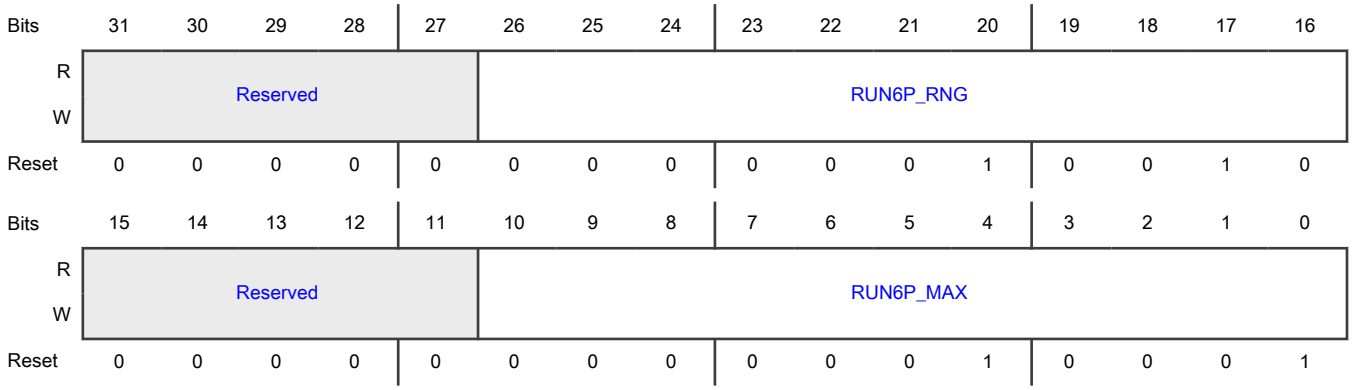
Function

The Statistical Check Run Length 6+ Limit Register defines the allowable maximum and minimum number of runs of length 6 or more detected during entropy generation. To pass the test, the number of runs of length 6 or more (for samples of both 0 and 1) must be less than the programmed maximum value, and the number of runs of length 6 or more must be greater than (RUN6P_MAX - RUN6P_RNG). If this test fails, the Retry Counter in SCMISC will be decremented, and a retry will occur if the Retry Count has not reached zero. If the Retry Count has reached zero, an error will be generated. Note that this offset (0x38) is used as SCR6PL only if MCTL[PRGM] is 1. If MCTL[PRGM] is 0, this offset is used as SCR6PC read back register.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram



Fields

Field	Function
31-27 —	Reserved
26-16 RUN6P_RNG	Run Length 6+ Range. The number of runs of length 6 or more (for both 0 and 1) detected during entropy generation must be greater than RUN6P_MAX - RUN6P_RNG, else a retry or error will occur. This register is cleared to the reset value by writing the MCTL[RST_DEF] bit to 1. Writing 1 to bit MCTL[DIS_SLF_TST] loads a value of 0x7FF in this field.
15-11 —	Reserved
10-0 RUN6P_MAX	Run Length 6+ Maximum Limit. Defines the maximum allowable runs of length 6 or more (for both 0 and 1) detected during entropy generation. The number of runs of length 6 or more detected during entropy generation must be less than RUN6P_MAX, else a retry or error will occur. This register is reset by writing the MCTL[RST_DEF] bit to 1. Writing 1 to bit MCTL[DIS_SLF_TST] loads a value of 0x7FE in this field.

41.7.1.28 Status Register (STATUS)

Offset

Register	Offset
STATUS	3Ch

Function

Various statistical tests are run as a normal part of the TRNG's entropy generation process. The least-significant 16 bits of the STATUS register reflect the result of each of these tests. The status of these bits will be valid when the TRNG has finished its entropy generation process. Software can determine when this occurs by polling MCTL[ENT_VAL].

Note that there is a very small probability that a statistical test will fail even though the TRNG is operating properly. If this happens the TRNG will automatically retry the entire entropy generation process, including running all the statistical tests. The value in RETRY_CT is decremented each time an entropy generation retry occurs. If a statistical check fails when the retry count is

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

nonzero, a retry is initiated. But if a statistical check fails when the retry count is zero, an error is generated by the TRNG. By default, RETRY_CT is initialized to 1, but software can increase the retry count by writing to SCMISC[RTY_CT].

All 0s will be returned if this register address is read while the TRNG is in Program Mode (MCTL[PRGM] = 1). If this register is read while the TRNG is in Run Mode (MCTL[PRGM] = 0), the value returned will be formatted as follows.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved												RETRY_CT			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TFMB	TFP	TFLR	TFSB	TF6PBR1	TF6PBR0	TF5BR1	TF5BR0	TF4BR1	TF4BR0	TF3BR1	TF3BR0	TF2BR1	TF2BR0	TF1BR1	TF1BR0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-20 —	Reserved
19-16 RETRY_CT	RETRY COUNT. This represents the current number of entropy generation retries left before a statistical test failure will cause the TRNG to generate an error condition.
15 TFMB	Test Fail, Mono Bit. 0b - The Mono Bit Test has passed 1b - The Mono Bit Test has failed
14 TFP	Test Fail, Poker. 0b - The Poker Test has passed 1b - The Poker Test has failed
13 TFLR	Test Fail, Long Run. 0b - The Long Run Test has passed 1b - The Long Run Test has failed
12 TFSB	Test Fail, Sparse Bit. 0b - The Sparse Bit Test has passed 1b - The Sparse Bit Test has failed
11	Test Fail, 6 Plus Bit Run, Sampling 1s.

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
TF6PBR1	0b - The 6 Plus Bit Run, Sampling 1s Test has passed 1b - The 6 Plus Bit Run, Sampling 1s Test has failed
10 TF6PBR0	Test Fail, 6 Plus Bit Run, Sampling 0s. 0b - The 6 Plus Bit Run, Sampling 0s Test has passed 1b - the 6 Plus Bit Run, Sampling 0s Test has failed
9 TF5BR1	Test Fail, 5-Bit Run, Sampling 1s. If TF5BR1=1, the 5-Bit Run, Sampling 1s Test has failed. 0b - The 5-Bit Run, Sampling 1s Test has passed 1b - The 5-Bit Run, Sampling 1s Test has failed
8 TF5BR0	Test Fail, 5-Bit Run, Sampling 0s. 0b - The 5-Bit Run, Sampling 0s Test has passed 1b - The 5-Bit Run, Sampling 0s Test has failed
7 TF4BR1	Test Fail, 4-Bit Run, Sampling 1s. 0b - The 4-Bit Run, Sampling 1s Test has passed 1b - The 4-Bit Run, Sampling 1s Test has failed
6 TF4BR0	Test Fail, 4-Bit Run, Sampling 0s 0b - The 4-Bit Run, Sampling 0s Test has passed 1b - The 4-Bit Run, Sampling 0s Test has failed
5 TF3BR1	Test Fail Test Fail, 3-Bit Run, Sampling 1s. 0b - The 3-Bit Run, Sampling 1s Test has passed 1b - The 3-Bit Run, Sampling 1s Test has failed
4 TF3BR0	Test Fail, 3-Bit Run, Sampling 0s. 0b - The 3-Bit Run, Sampling 0s Test has passed 1b - The 3-Bit Run, Sampling 0s Test has failed
3 TF2BR1	Test Fail, 2-Bit Run, Sampling 1s. 0b - The 2-Bit Run, Sampling 1s Test has passed 1b - The 2-Bit Run, Sampling 1s Test has failed
2 TF2BR0	Test Fail, 2-Bit Run, Sampling 0s. 0b - The 2-Bit Run, Sampling 0s Test has passed 1b - The 2-Bit Run, Sampling 0s Test has failed

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
1 TF1BR1	Test Fail, 1-Bit Run, Sampling 1s. 0b - The 1-Bit Run, Sampling 1s Test has passed 1b - The 1-Bit Run, Sampling 1s Test has failed
0 TF1BR0	Test Fail, 1-Bit Run, Sampling 0s. 0b - The 1-Bit Run, Sampling 0s Test has passed 1b - The 1-Bit Run, Sampling 0s Test has failed

41.7.1.29 Entropy Read Register (ENT0 - ENT7)

Offset

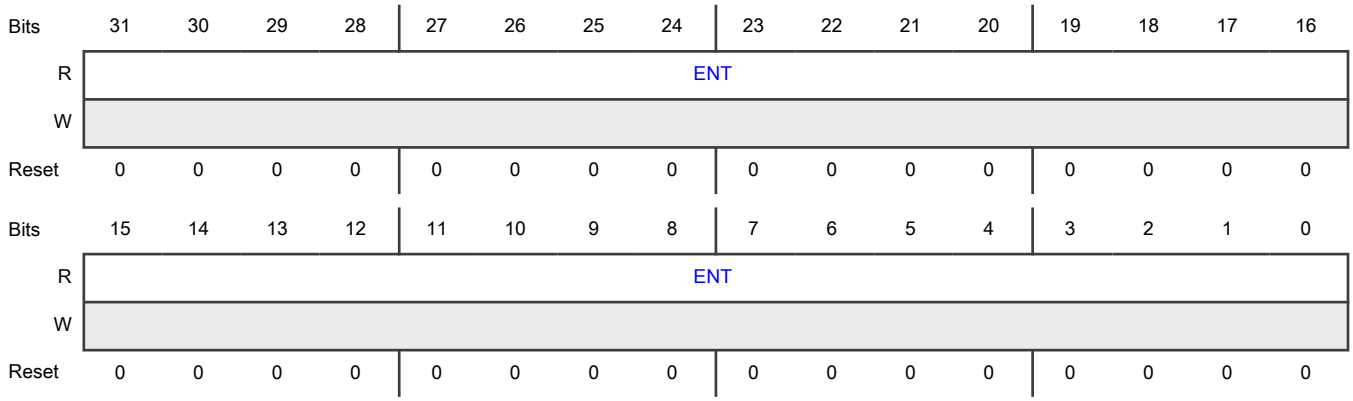
For a = 0 to 7:

Register	Offset	Description
ENTa	40h + (a × 4h)	Word a

Function

The TRNG can be programmed to generate an entropy value that is readable via the register interface. To do this, set the MCTL[TRNG_ACC] bit to 1. Once the entropy value has been generated, the MCTL[ENT_VAL] bit will be set to 1. At this point, ENT0 through ENT7 may be read to retrieve the 256-bit entropy value. Note that once ENT7 is read, the entropy value will be cleared and a new value will begin generation, so it is important that ENT7 be read last. These registers are readable only when MCTL[PRGM] = 0 (Run Mode) , MCTL[TRNG_ACC] = 1 (TRNG access mode) and MCTL[ENT_VAL] = 1. After at most one (1) bus clock cycle of reading a valid ENT7 register value, reading any ENT0 through ENT7 register will return zeros.

Diagram



All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Fields

Field	Function
31-0 ENT	Entropy Value Reads as zero except when MCTL[PRGM] = 0 (Run Mode) and MCTL[ENT_VAL] = 1 (Entropy Valid). Reading the entropy value at the highest offset clears all entropy values, and starts a new entropy generation.

41.7.1.30 Statistical Check Poker Count 1 and 0 Register (PKRCNT10)

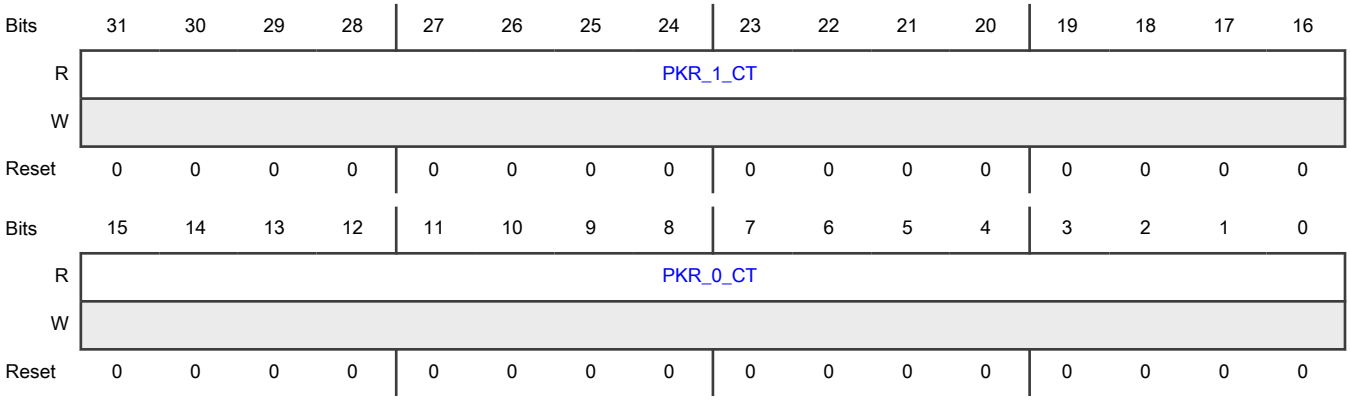
Offset

Register	Offset
PKRCNT10	80h

Function

The Statistical Check Poker Count 1 and 0 Register is a read-only register used to read the final Poker test counts of 1h and 0h patterns. The Poker 0h Count increments each time a nibble of sample data is found to be 0h. The Poker 1h Count increments each time a nibble of sample data is found to be 1h. Note that this register is readable only if MCTL[PRGM] is 0, otherwise zeros will be read.

Diagram



Fields

Field	Function
31-16 PKR_1_CT	Poker 1h Count. Total number of nibbles of sample data which were found to be 1h. Requires MCTL[PRGM] = 0.
15-0 PKR_0_CT	Poker 0h Count. Total number of nibbles of sample data which were found to be 0h. Requires MCTL[PRGM] = 0.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

41.7.1.31 Statistical Check Poker Count 3 and 2 Register (PKRCNT32)

Offset

Register	Offset
PKRCNT32	84h

Function

The Statistical Check Poker Count 3 and 2 Register is a read-only register used to read the final Poker test counts of 3h and 2h patterns. The Poker 2h Count increments each time a nibble of sample data is found to be 2h. The Poker 3h Count increments each time a nibble of sample data is found to be 3h. Note that this register is readable only if MCTL[PRGM] is 0, otherwise zeros will be read.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	PKR_3_CT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PKR_2_CT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-16 PKR_3_CT	Poker 3h Count. Total number of nibbles of sample data which were found to be 3h. Requires MCTL[PRGM] = 0.
15-0 PKR_2_CT	Poker 2h Count. Total number of nibbles of sample data which were found to be 2h. Requires MCTL[PRGM] = 0.

41.7.1.32 Statistical Check Poker Count 5 and 4 Register (PKRCNT54)

Offset

Register	Offset
PKRCNT54	88h

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Function

The Statistical Check Poker Count 5 and 4 Register is a read-only register used to read the final Poker test counts of 5h and 4h patterns. The Poker 4h Count increments each time a nibble of sample data is found to be 4h. The Poker 5h Count increments each time a nibble of sample data is found to be 5h. Note that this register is readable only if MCTL[PRGM] is 0, otherwise zeros will be read.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	PKR_5_CT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PKR_4_CT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-16 PKR_5_CT	Poker 5h Count. Total number of nibbles of sample data which were found to be 5h. Requires MCTL[PRGM] = 0.
15-0 PKR_4_CT	Poker 4h Count. Total number of nibbles of sample data which were found to be 4h. Requires MCTL[PRGM] = 0.

41.7.1.33 Statistical Check Poker Count 7 and 6 Register (PKRCNT76)

Offset

Register	Offset
PKRCNT76	8Ch

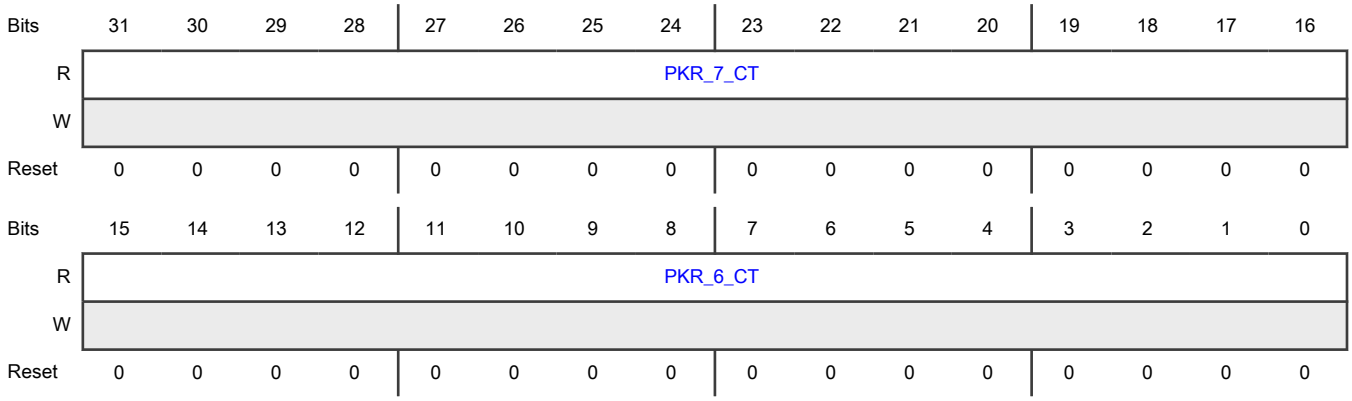
Function

The Statistical Check Poker Count 7 and 6 Register is a read-only register used to read the final Poker test counts of 7h and 6h patterns. The Poker 6h Count increments each time a nibble of sample data is found to be 6h. The Poker 7h Count increments each time a nibble of sample data is found to be 7h. Note that this register is readable only if MCTL[PRGM] is 0, otherwise zeros will be read.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram



Fields

Field	Function
31-16 PKR_7_CT	Poker 7h Count. Total number of nibbles of sample data which were found to be 7h. Requires MCTL[PRGM] = 0.
15-0 PKR_6_CT	Poker 6h Count. Total number of nibbles of sample data which were found to be 6h. Requires MCTL[PRGM] = 0.

41.7.1.34 Statistical Check Poker Count 9 and 8 Register (PKRCNT98)

Offset

Register	Offset
PKRCNT98	90h

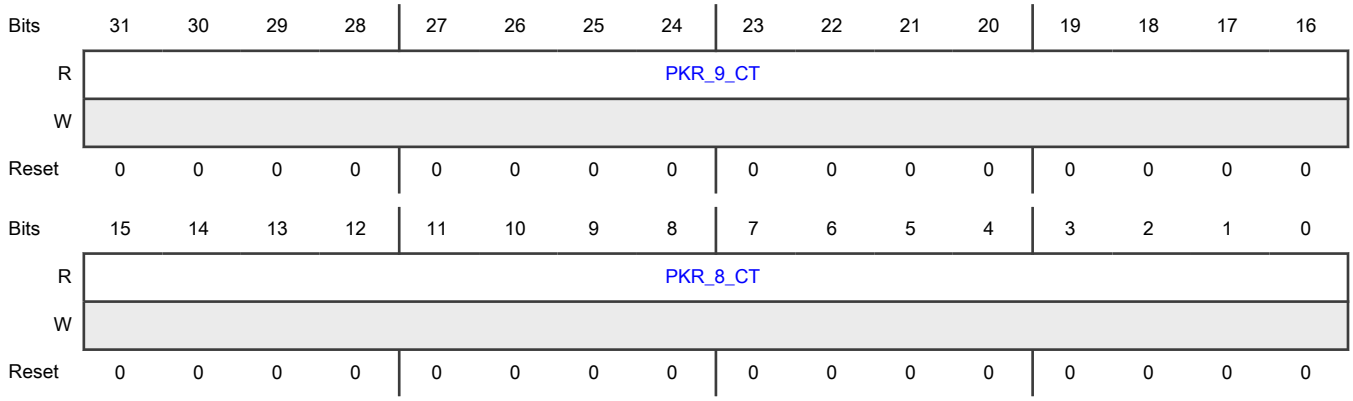
Function

The Statistical Check Poker Count 9 and 8 Register is a read-only register used to read the final Poker test counts of 9h and 8h patterns. The Poker 8h Count increments each time a nibble of sample data is found to be 8h. The Poker 9h Count increments each time a nibble of sample data is found to be 9h. Note that this register is readable only if MCTL[PRGM] is 0, otherwise zeros will be read.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram



Fields

Field	Function
31-16 PKR_9_CT	Poker 9h Count Total number of nibbles of sample data which were found to be 9h. Requires MCTL[PRGM] = 0.
15-0 PKR_8_CT	Poker 8h Count Total number of nibbles of sample data which were found to be 8h. Requires MCTL[PRGM] = 0.

41.7.1.35 Statistical Check Poker Count B and A Register (PKRCNTBA)

Offset

Register	Offset
PKRCNTBA	94h

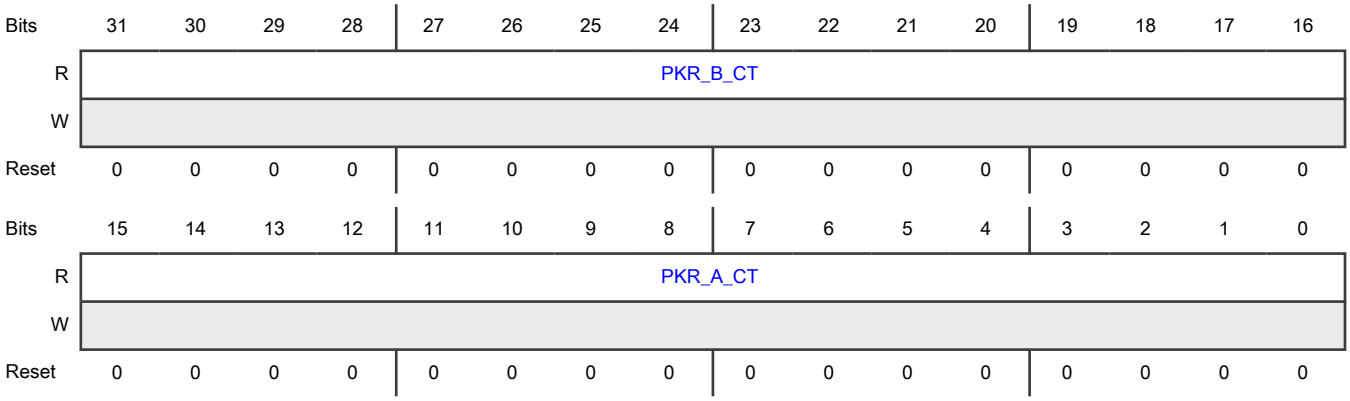
Function

The Statistical Check Poker Count B and A Register is a read-only register used to read the final Poker test counts of Bh and Ah patterns. The Poker Ah Count increments each time a nibble of sample data is found to be Ah. The Poker Bh Count increments each time a nibble of sample data is found to be Bh. Note that this register is readable only if MCTL[PRGM] is 0, otherwise zeros will be read.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram



Fields

Field	Function
31-16 PKR_B_CT	Poker Bh Count Total number of nibbles of sample data which were found to be Bh. Requires MCTL[PRGM] = 0.
15-0 PKR_A_CT	Poker Ah Count Total number of nibbles of sample data which were found to be Ah. Requires MCTL[PRGM] = 0.

41.7.1.36 Statistical Check Poker Count D and C Register (PKRCNTDC)

Offset

Register	Offset
PKRCNTDC	98h

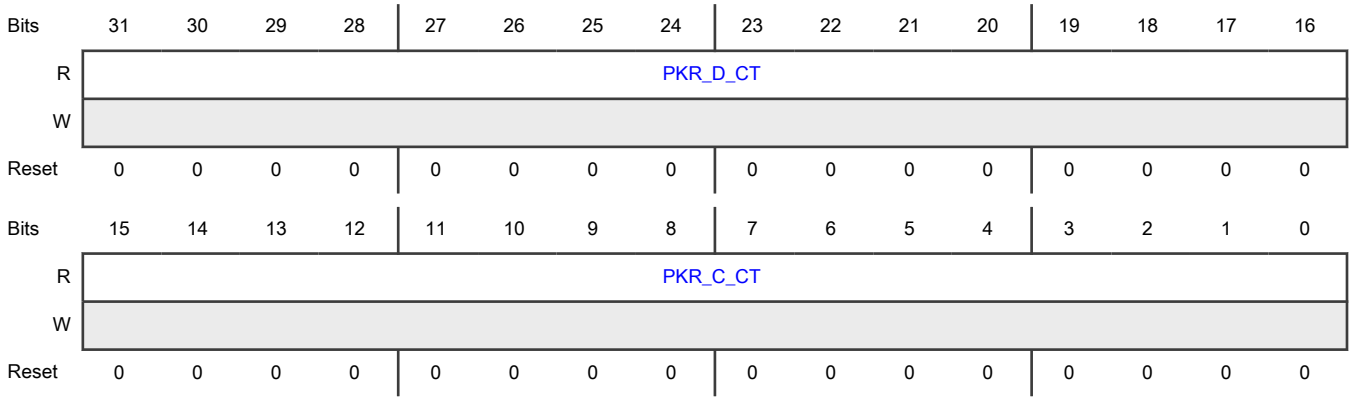
Function

The Statistical Check Poker Count D and C Register is a read-only register used to read the final Poker test counts of Dh and Ch patterns. The Poker Ch Count increments each time a nibble of sample data is found to be Ch. The Poker Dh Count increments each time a nibble of sample data is found to be Dh. Note that this register is readable only if MCTL[PRGM] is 0, otherwise zeros will be read.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram



Fields

Field	Function
31-16 PKR_D_CT	Poker Dh Count Total number of nibbles of sample data which were found to be Dh. Requires MCTL[PRGM] = 0.
15-0 PKR_C_CT	Poker Ch Count Total number of nibbles of sample data which were found to be Ch. Requires MCTL[PRGM] = 0.

41.7.1.37 Statistical Check Poker Count F and E Register (PKRCNTFE)

Offset

Register	Offset
PKRCNTFE	9Ch

Function

The Statistical Check Poker Count F and E Register is a read-only register used to read the final Poker test counts of Fh and Eh patterns. The Poker Eh Count increments each time a nibble of sample data is found to be Eh. The Poker Fh Count increments each time a nibble of sample data is found to be Fh. Note that this register is readable only if MCTL[PRGM] is 0, otherwise zeros will be read.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	PKR_F_CT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PKR_E_CT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-16 PKR_F_CT	Poker Fh Count. Total number of nibbles of sample data which were found to be Fh. Requires MCTL[PRGM] = 0.
15-0 PKR_E_CT	Poker Eh Count. Total number of nibbles of sample data which were found to be Eh. Requires MCTL[PRGM] = 0.

41.7.1.38 Security Configuration Register (SEC_CFG)

Offset

Register	Offset
SEC_CFG	A0h

Function

The Security Configuration Register used to control global configuration of TRNG.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved														NO_PRGM	Reserved
W															1	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Fields

Field	Function
31-2 —	Reserved
1 NO_PRGM	If set, below mentioned TRNG configuration registers cannot be programmed: • Oscillator 2 Control Register (OSC2_CTL): — TRNG Entropy Generation Control [1:0] — Oscillator 2 Divider [3:2] — Oscillator Fail Safe Limit [13:12] — Oscillator Fail Safe Test [14] • TRNG Seed Control Register (SDCTL) • TRNG Frequency Count Minimum Limit Register (FRQMIN) • TRNG Frequency Count Maximum Limit Register (FRQMAX) • TRNG Statistical Check Monobit Limit Register (SCML) • TRNG Statistical Check Run Length 1 Limit Register (SCR1L) • TRNG Statistical Check Run Length 2 Limit Register (SCR2L) • TRNG Statistical Check Run Length 3 Limit Register (SCR3L) • TRNG Miscellaneous Control Register (MCTL): — Sample Mode [1:0] — Oscillator Divider [3:2] — Reset Defaults [6] — Force System Clock [7] — Long Runs Continuation Mode [14] After this bit has been written to a 1, it cannot be changed. The bit is reset to 0 by an asynchronous reset. 0b - TRNG configuration registers can be modified. 1b - TRNG configuration registers cannot be modified.
0 —	Reserved

41.7.1.39 Interrupt Control Register (INT_CTRL)

Offset

Register	Offset
INT_CTRL	A4h

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Function

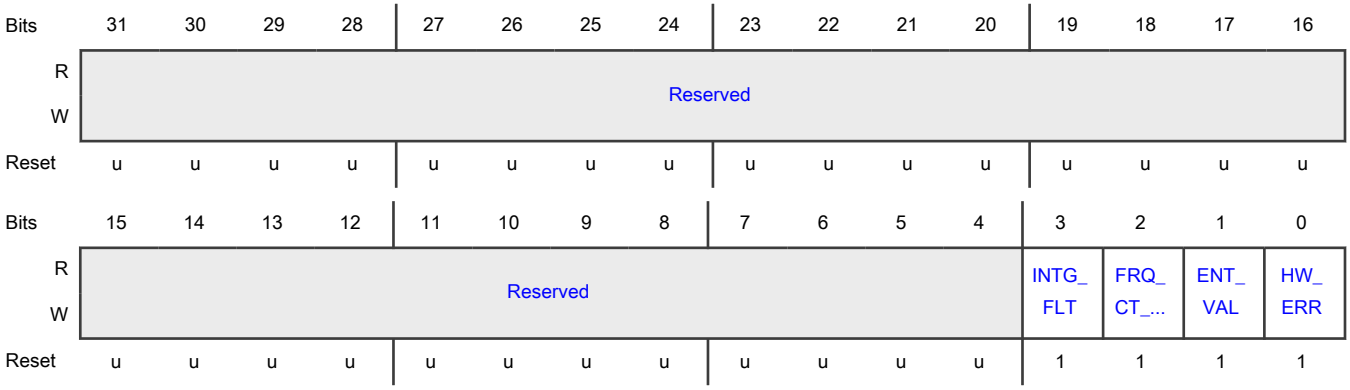
The Interrupt Control Register is a read/write register used to clear the status for the four interrupts generated by the TRNG. See INT_STATUS register description.

The purpose of the INT_CTRL register is to allow a pending interrupt to be cleared. To perform this function the specific interrupt bit is written 0. At this time the INT_STATUS bit for the corresponding interrupt, and the interrupt output (if no other interrupts were pending) will de-assert.

The INT_CTRL bit for the interrupt will remain 0 as long as the interrupt source that caused the interrupt remains asserted. Once the source is de-asserted the INT_CTRL bit will return 1 and remain there for the next interrupt.

Writing 1 can NOT cause generation of interrupts, or have any other effect.

Diagram



Fields

Field	Function
31-4 —	Reserved
3 INTG_FLT	Clear the INTG_FLT interrupt. 0b - Clears the INT_STATUS[INTG_FLT] bit. Will automatically set after writing. 1b - Enables the INT_STATUS[INTG_FLT] bit to be set, thereby enabling interrupt generation for the INTG_FLT condition.
2 FRQ_CT_FAIL	Clear the FRQ_CT_FAIL interrupt. 0b - Clears the INT_STATUS[FRQ_CT_FAIL] bit. Will automatically set after writing. 1b - Enables the INT_STATUS[FRQ_CT_FAIL] bit to be set, thereby enabling interrupt generation for the FRQ_CT_FAIL condition.
1 ENT_VAL	Clear the ENT_VAL interrupt. 0b - Clears the INT_STATUS[ENT_VAL] bit. Will automatically set after writing. 1b - Enables the INT_STATUS[ENT_VAL] bit to be set, thereby enabling interrupt generation for the ENT_VAL condition.
0	Clear the HW_ERR interrupt.

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
HW_ERR	0b - Clears the INT_STATUS[HW_ERR] bit. Will automatically set after writing. 1b - Enables the INT_STATUS[HW_ERR] bit to be set, thereby enabling interrupt generation for the HW_ERR condition.

41.7.1.40 Mask Register (INT_MASK)

Offset

Register	Offset
INT_MASK	A8h

Function

The Interrupt Mask Register (actually an enable register) is a read/write register used to enable the status reporting of the four interrupts generated by the TRNG. See INT_STATUS register description. Each interrupt can be masked/disabled by clearing the corresponding bit in the INT_MASK register. Only setting this bit high will re-enable the interrupt in the status register. Even if the interrupt is cleared or masked, interrupt status information can be read from the MCTL register.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved												INTG_FLT	FRQ_CT_...	ENT_VAL	HW_ERR
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fields

Field	Function
31-4 —	Reserved
3 INTG_FLT	Mask the INTG_FLT interrupt. 0b - INTG_FLT interrupt is disabled. 1b - INTG_FLT interrupt is enabled.

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
2 FRQ_CT_FAIL	Mask the FRQ_CT_FAIL interrupt. 0b - FRQ_CT_FAIL interrupt is disabled. 1b - FRQ_CT_FAIL interrupt is enabled.
1 ENT_VAL	Mask the ENT_VAL interrupt. 0b - ENT_VAL interrupt is disabled. 1b - ENT_VAL interrupt is enabled.
0 HW_ERR	Mask the HW_ERR interrupt. 0b - HW_ERR interrupt is disabled. 1b - HW_ERR interrupt is enabled.

41.7.1.41 Interrupt Status Register (INT_STATUS)

Offset

Register	Offset
INT_STATUS	ACh

Function

The Interrupt Status Register is used to provide status for the three interrupts generated by the TRNG. These interrupts can be used to provide alert when either Frequency Count has failed or Entropy is Valid or an HW Error has occurred. Each interrupt can be temporarily cleared by writing 0 to the corresponding bit in the INT_CTRL register. To mask the interrupts, write 0 to the corresponding bits in the INT_MASK register. The description of the interrupts is defined in the MCTL register description. Even if the interrupt is cleared or masked, interrupt status information can be read from the MCTL register.

Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved												INTG_FLT	FRQ_CT_...	ENT_VAL	HW_ERR
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Fields

Field	Function
31-4 —	Reserved
3 INTG_FLT	Integrity Fault. An internal fault has occurred. 0b - No internal fault has been detected. 1b - TRNG has detected internal fault.
2 FRQ_CT_FAIL	Frequency Count Fail. The frequency counter has detected a failure. This may be due to improper programming of the FRQMAX and/or FRQMIN registers, or a hardware failure in the ring oscillator. 0b - No hardware nor self test frequency errors. 1b - The frequency counter has detected a failure.
1 ENT_VAL	Entropy Valid. Will assert after an entropy value is generated and only if TRNG_ACC bit is set. Will be cleared when ENT7 is read. (ENT0 through ENT6 should be read before reading ENT7). 0b - Busy generating entropy. Any value read from the Entropy registers is invalid. 1b - Values read from the Entropy registers are valid.
0 HW_ERR	TRNG Error. Any error in the TRNG will trigger this interrupt. 0b - No error. 1b - Error detected.

41.7.1.42 TRNG Oscillator 2 Control Register (OSC2_CTL)

Offset

Register	Offset
OSC2_CTL	ECh

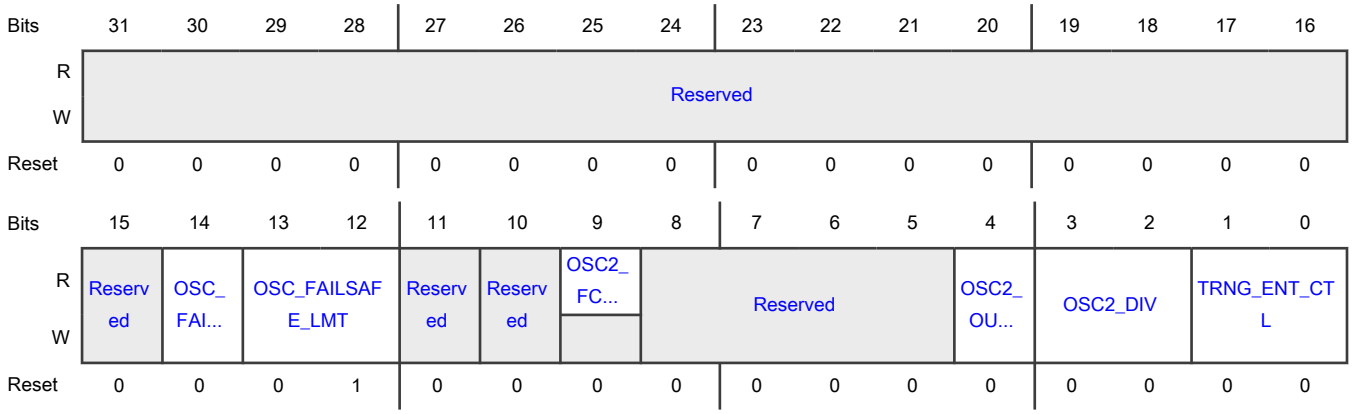
Function

The TRNG Oscillator 2 Control Register controls various functions related to the second free-running oscillator in the TRNG. OSC2_CTL is only write-able if MCTL[PRGM]=1.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram



Fields

Field	Function
31-15 —	Reserved
14 OSC_FAILSAFE_TEST	Oscillator fail safe test. This is used only for testing the OSC_FAILSAFE_LMT logic. This field is writable only if MCTL[PRGM] bit is 1. This field is reset by writing the RST_DEF bit to 1. 0b - No impact. 1b - Disables oscillator 2 while in dual-oscillator mode (TRNG_ENT_CTL = 01b).
13-12 OSC_FAILSAFE_LMT	Oscillator fail safe limit. Sets the fail-safe limit for determining oscillator failure. This field is writable only if MCTL[PRGM] bit is 1. This field is reset by writing the RST_DEF bit to 1. In dual-oscillator mode (TRNG_ENT_CTL = 01b), if oscillator 2 is not running, both MCTL[FCT_FAIL] and MCTL[OSC2_FAIL] will be set after N system clocks, as follows: 00b - The limit N is 4096 (2^12) system clocks. 01b - The limit N is 65536 (2^16) system clocks. (default) 10b - N is 2^20 system clocks. 11b - N is 2^22 system clocks (full range of the counter being used).
11 —	Reserved
10 —	Reserved
9 OSC2_FCT_VAL	TRNG Oscillator 2 Frequency Count Valid 0b - Frequency count is invalid. 1b - If TRNG_ENT_CTL = 10b, valid frequency count may be read from OSC2_FRQCNT.

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Table continued from the previous page...

Field	Function
8-5 —	Reserved
4 OSC2_OUT_EN	Oscillator 2 Clock Output Enable This field is reset by writing the RST_DEF bit to 1. 0b - Ring oscillator 2 output is gated to an output pad. 1b - Allows external viewing of divided-by-2 ring oscillator 2 if MCTL[PRGM] = 1 mode is also selected, else ring oscillator 2 output is gated to an output pad.
3-2 OSC2_DIV	Oscillator 2 Divide. Determines the amount of dividing done to the ring oscillator 2 before it is used by the TRNG. This field is writable only if MCTL[PRGM] is 1. This field is reset by writing the RST_DEF bit to 1. 00b - Use ring oscillator 2 with no divide 01b - Use ring oscillator 2 divided-by-2 10b - Use ring oscillator 2 divided-by-4 11b - Use ring oscillator 2 divided-by-8
1-0 TRNG_ENT_CTL	TRNG entropy generation control. This field is writable only if MCTL[PRGM] is 1. This field is reset by writing the RST_DEF bit to 1. 00b - Single oscillator mode, using OSC1 (default) 01b - Dual oscillator mode 10b - Single oscillator mode, using OSC2 11b - Unused, (bit field cannot be written to this value)

41.7.1.43 Version ID Register (MS) (VID1)

Offset

Register	Offset
VID1	F0h

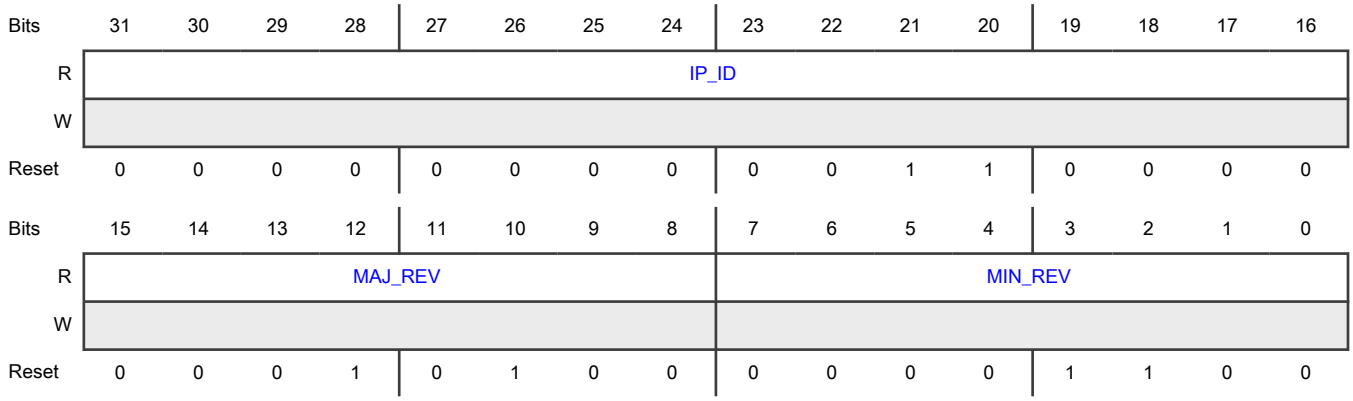
Function

The Version ID Register (VID1) is an MSW read only register used to identify the version of the TRNG in use. This register as well as VID2 should both be read to verify the expected version.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram



Fields

Field	Function
31-16 IP_ID	Shows the IP ID. 0000_0000_0011_0000b - ID for TRNG.
15-8 MAJ_REV	Shows the IP's Major revision of the TRNG. • A value of 10000000b indicates a MTRNG wrapper. • A value of 00010000b indicates TRNG has Dual-Oscillators. • A value of 00000100b indicates TRNG with 256-bit entropy size. • A value of 00000010b indicates TRNG with 128-bit entropy size. • A value of 00000001b indicates TRNG with 512-bit entropy size. 0001_0100b - Major revision number for TRNG.
7-0 MIN_REV	Shows the IP's Minor revision of the TRNG. 0000_1100b - Minor revision number for TRNG.

41.7.1.44 Version ID Register (LS) (VID2)

Offset

Register	Offset
VID2	F4h

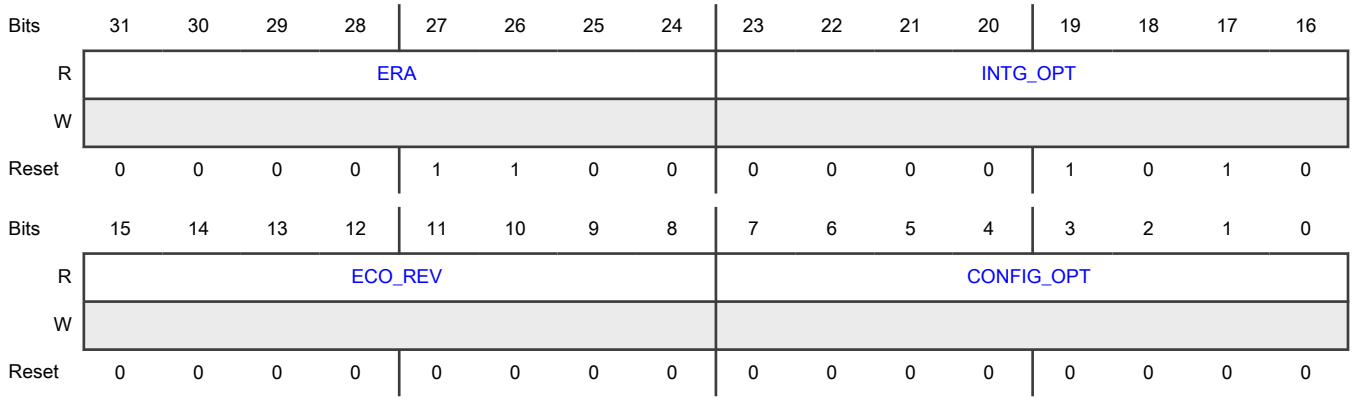
Function

The Version ID Register (VID2) is an LSW read only register used to identify the architecture of the TRNG in use. This register as well as VID1 should both be read to verify the expected version.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Diagram



Fields

Field	Function
31-24 ERA	Shows the ERA of the TRNG. 0000_1100b - ERA of the TRNG.
23-16 INTG_OPT	Shows the integration options for the TRNG. 0000_1010b - INTG_OPT for TRNG.
15-8 ECO_REV	Shows the IP's ECO revision of the TRNG. 0000_0000b - TRNG_ECO_REV for TRNG.
7-0 CONFIG_OPT	Shows the IP's Configuration options for the TRNG. 0000_0000b - TRNG_CONFIG_OPT for TRNG.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Appendix A
Release notes

A.1 About This Manual changes

No substantial content changes.

A.2 Introduction changes

- Updated Memory features and packages information in Features.

A.3 Core overview changes

- Updated System memory map.

A.4 Bus and memory architecture chapter changes

- Added IFR configuration.

A.5 Flash Memory Module (FMU)

A.5.1 Chip-specific FMU information changes

No substantial content changes.

A.5.2 FMU changes

No substantial content changes.

A.6 Flash Memory Controller (FMC)

A.6.1 Chip-specific FMC information changes

No substantial content changes.

A.6.2 Flash Memory Controller (FMC-NPX) module changes

No substantial content changes.

A.7 Error Injection Module (EIM)

A.7.1 Chip-specific EIM information changes

No substantial content changes.

All information provided in this document is subject to legal disclaimers.
© 2026 NXP B.V. All rights reserved.

A.7.2 EIM module changes

No substantial content changes.

A.8 Error Recording Module (ERM)

A.8.1 Chip-specific ERM information changes

No substantial content changes.

A.8.2 ERM module changes

No substantial content changes.

A.9 Signal multiplexing chapter changes

- Updated package information in Signal multiplexing and pin assignments and attached pinout table.

A.10 Port Control (PORT)

A.10.1 Chip-specific PORT information changes

No substantial content changes.

A.10.2 PORT module changes

- Added a Note to the description for PCR*n*[IBE].

A.11 General Purpose Input/Output (GPIO)

A.11.1 Chip-specific GPIO information changes

No substantial content changes.

A.11.2 GPIO module changes

No substantial content changes.

A.12 Input Multiplexing (INPUTMUX)

A.12.1 Chip-specific INPUTMUX information changes

No substantial content changes.

A.12.2 INPUTMUX module changes

No substantial content changes.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

A.13 System Controller (SYSCON)

A.13.1 Chip-specific SYSCON information changes

No substantial content changes.

A.13.2 SYSCON module changes

- Initial Draft
- Updated Pins that have CLKOUT function in Table 74.
- Updated Pins that have CLKOUT function in Table 74.

A.14 Enhanced DMA Controller (eDMA3)

A.14.1 Chip-specific eDMA3 information changes

No substantial content changes.

A.14.2 eDMA3 module changes

- In CHn_CSR[DONE] , added a note "When an error is injected into the EDMA TCD---may not assert the DONE signal".

A.15 Wakeup Unit (WUU)

A.15.1 Chip-specific WUU information changes

No substantial content changes.

A.15.2 WUU changes

- Updated Power Down mode

A.16 Core Mode Controller (CMC)

A.16.1 Chip-specific CMC information changes

No substantial content changes.

A.16.2 CMC module changes

- Added .
- Added .
- Updated SRIF[23] as Reserved.

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

- Added SRIF[TAMPER].
- Changed the bit value of PMCTRLMAIN[LPMODE]

A.17 Reset changes

No substantial content changes.

A.18 Lifecycle States changes

- Updated Customer lifecycle states.
- Added a note in Lifecycle states transition from Develop to in-field.
- Updated Develop lifecycle state.
- Updated Table 95.

A.19 Bootloader changes

- Updated Boot image.
- Updated Overview.
- Added Debug session with Bootloader ISP mode entered to replace Debug session with valid application.
- Updated Debug session with uninitialized or invalid image.
- Updated Debug session usage instructions.
- Updated Debug session attaching to a running target.
- Updated Flash-erase-all.
- Updated MISR image.
- Updated DM-AP commands.
- Added Low-power wakeup path.
- Updated the default operating frequency of boot flow to 48 MHz in Boot flow.
- Updated CMPA configuration options.
- Added Available properties.
- Updated Table 136.
- Updated CMPA configuration options.

A.20 Clocking chapter changes

No substantial content changes.

A.21 System Clock Generator (SCG)

A.21.1 Chip-specific SCG information changes

No substantial content changes.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

A.21.2 SCG module changes

• Updated the following register function descriptions: — SIRC Auto-trimming Status (SIRCSTAT) — SIRC Trim (SIRCTRIM) — FIRC Trim (FIRCTRIM)
• Updated the bitfield value description of the following fields: — CSR[SCS] — RCCR[SCS]
• Updated Resets and Initialization

A.22 Power Management chapter changes

No substantial content changes.

A.23 VBAT

A.23.1 Chip-specific VBAT information changes

No substantial content changes.

A.23.2 VBAT module changes

No substantial content changes.

A.24 System Power Control (SPC)

A.24.1 Chip-specific SPC information changes

No substantial content changes.

A.24.2 SPC module changes

• Changed reset value of bit 12 to 1 in Low-Power Mode Configuration (LP_CFG)
• Fixed some typo errors • Updated ACTIVE_CFG[CORELDO_VDD_LVL] and LP_CF[CORELDO_VDD_LVL] bit voltage for internal use
• Updated the value of SRAMCTL[VSM].

A.25 Standard Counter/Timers (CTIMER)

A.25.1 Chip-specific CTIMER information changes

No substantial content changes.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

A.25.2 CTIMER module changes

No substantial content changes.

A.26 Windowed Watchdog Timer (WWDT)

A.26.1 Chip-specific WWDT information changes

No substantial content changes.

A.26.2 WWDT module changes

No substantial content changes.

A.27 Real Time Clock (RTC)

A.27.1 Chip-specific RTC information changes

No substantial content changes.

A.27.2 RTC module changes

No substantial content changes.

A.28 Enhanced Flex Pulse Width Modulator (eFlexPWM)

A.28.1 Chip-specific eFlexPWM information changes

No substantial content changes.

A.28.2 eFlexPWM module changes

No substantial content changes.

A.29 Frequency Measurement (FREQME)

A.29.1 Chip-specific FREQME information changes

No substantial content changes.

A.29.2 FREQME module changes

No substantial content changes.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

A.30 Low-power Timer (LPTMR)

A.30.1 Chip-specific LPTMR information changes

No substantial content changes.

A.30.2 LPTMR module changes

No substantial content changes.

A.31 Low Power Inter-Integrated Circuit (LPI2C)

A.31.1 Chip-specific LPI2C information changes

No substantial content changes.

A.31.2 LPI2C module changes

- Updated Features Arbitration list item.
- Updated Clock stretching and SCFGR2[CLKHOLD] for disabling clock stretching extension.

A.32 Low Power Universal Asynchronous Receiver/Transmitter (LPUART)

A.32.1 Chip-specific LPUART information changes

No substantial content changes.

A.32.2 LPUART module changes

- In RXUF, added note.

A.33 Low Power Serial Peripheral Interface (LPSPI)

A.33.1 Chip-specific LPSPI information changes

No substantial content changes.

A.33.2 LPSPI module changes

- Updated Timing parameters, corrected register name for CCR[PCSSCK] and CCR[SCKPCS].

A.34 Analog-to-Digital Converter (ADC)

A.34.1 Chip-specific ADC information changes

No substantial content changes.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

A.34.2 ADC module changes

No substantial content changes.

A.35 Operational Amplifier (OPAMP)

A.35.1 Chip-specific OPAMP information changes

• Updated the input of OPAMP_CTR[PREF] in Figure 145. • Updated Figure 145.

A.35.2 OPAMP module changes

• Updated bitfield value definitions in OPAMP_CTR[BIASC]
• Updated Block diagram • Updated the field description of OPAMP_CTR[PREF]

A.36 Low Power Comparator (LPCMP)

A.36.1 Chip-specific LPCMP information changes

No substantial content changes.

A.36.2 LPCMP module changes

No substantial content changes.

A.37 Debug chapter changes

No substantial content changes.

A.38 DBGMB module changes

No substantial content changes.

A.39 Cyclic Redundancy Check (CRC)

A.39.1 Chip-specific CRC information changes

No substantial content changes.

A.39.2 CRC module changes

• Removed the note in Clocking
• Added details about CTRL[TOT] transpose effect on SEED in note of Types of transpose

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

A.40 Memory Block Checker (MBC)

A.40.1 Chip-specific MBC information changes

No substantial content changes.

A.40.2 MBC module changes

• Updated description of NSE and END_ADDR bits.
• Fixed reset value issue of register DACFG_Ww.

A.41 True Random Generator (TRNG)

A.41.1 TRNG bridge topic changes

No substantial content changes.

A.41.2 TRNG module changes

• Updated following sections: — Updated Block Diagram — Updated — Updated Functional Description — Updated Operation — Updated Interrupt Programming Flow — Updated Standalone Operation Programming Flow — Removed "2500 [SAMP_SIZE]" column in Table 246 — Updated "SP800-90" to "SP800-90A" in Other Applications • Updated description for following registers: — Interrupt Status Register (INT_STATUS) — Status Register (STATUS) • Updated descriptions for following fields: — MCTL[FCT_FAIL] — MCTL[TSTOP_OK] — MCTL[LRUN_CONT] — MCTL[PRGM] — SDCTL[ENT_DLY] — SDCTL[SAMP_SIZE] — VID2[INTG_OPT]

Table continues on the next page...

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

  - VID1[MAJ_REV]
  - INT_CTRL[FRQ_CT_FAIL]
  - INT_CTRL[ENT_VAL]
  - INT_CTRL[ENT_VAL]
  - SEC_CFG[NO_PRGM]
- Updated following register fields as Reserved :
  - MCTL[4-5]
  - MCTL[7]
  - MCTL[11]
  - OSC2_CTL[11]
  - OSC2_CTL[4]
- Updated "cleared to the POR reset value by", "POR reset value", "default POR reset value, "POR default reset value" references to "reset"
- Updated "DTRNG", "RNG" references to "TRNG"
- Updated "Oscillator Divide" to "Oscillator1 Divide" in MCTL[OSC_DIV] field description
- Updated Reserved field descriptions from "Reserved Always 0" to "Reserved"
- Updated "(maximum - range)" to "MONO_MAX minus MONO_RNG" in SCML register description
- Updated "(maximum - range)" to "RUN1_MAX minus RUN1_RNG" in SCR1L register description
- Updated "(maximum - range)" to "RUN2_MAX - RUN2_RNG" in SCR2L register description
- Updated "(maximum - range)" to "RUN3_MAX - RUN3_RNG" in SCR3L register description

- Added text "Setting the MCTL[PRGM] bit will also cause TSTOP_OK to assert" to the "MCTL[TSTOP_OK]" bit description.
- Added text "Writing 1 to bit MCTL[DIS_SLF_TST] loads a value of 0xFF in this field" for register fields affected by writing 1 to bit MCTL[DIS_SLF_TST].

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Legal information

Definitions

Draft — A draft status on a document indicates that the content is still under internal review and subject to formal approval, which may result in modifications or additions. NXP Semiconductors does not give any representations or warranties as to the accuracy or completeness of information included in a draft version of a document and shall have no liability for the consequences of use of such information.

Disclaimers

Limited warranty and liability — Information in this document is believed to be accurate and reliable. However, NXP Semiconductors does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information. NXP Semiconductors takes no responsibility for the content in this document if provided by an information source outside of NXP Semiconductors.

In no event shall NXP Semiconductors be liable for any indirect, incidental, punitive, special or consequential damages (including - without limitation - lost profits, lost savings, business interruption, costs related to the removal or replacement of any products or rework charges) whether or not such damages are based on tort (including negligence), warranty, breach of contract or any other legal theory.

Notwithstanding any damages that customer might incur for any reason whatsoever, NXP Semiconductors' aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the Terms and conditions of commercial sale of NXP Semiconductors.

Right to make changes — NXP Semiconductors reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions, at any time and without notice. This document supersedes and replaces all information supplied prior to the publication hereof.

Suitability for use — NXP Semiconductors products are not designed, authorized or warranted to be suitable for use in life support, life-critical or safety-critical systems or equipment, nor in applications where failure or malfunction of an NXP Semiconductors product can reasonably be expected to result in personal injury, death or severe property or environmental damage. NXP Semiconductors and its suppliers accept no liability for inclusion and/or use of NXP Semiconductors products in such equipment or applications and therefore such inclusion and/or use is at the customer's own risk.

Applications — Applications that are described herein for any of these products are for illustrative purposes only. NXP Semiconductors makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

Customers are responsible for the design and operation of their applications and products using NXP Semiconductors products, and NXP Semiconductors accepts no liability for any assistance with applications or customer product design. It is customer's sole responsibility to determine whether the NXP Semiconductors product is suitable and fit for the customer's applications and products planned, as well as for the planned application and use of customer's third party customer(s). Customers should provide appropriate design and operating safeguards to minimize the risks associated with their applications and products.

NXP Semiconductors does not accept any liability related to any default, damage, costs or problem which is based on any weakness or default in the customer's applications or products, or the application or use by customer's third party customer(s). Customer is responsible for doing all necessary testing for the customer's applications and products using NXP Semiconductors products in order to avoid a default of the applications and the products or of the application or use by customer's third party customer(s). NXP does not accept any liability in this respect.

Terms and conditions of commercial sale — NXP Semiconductors products are sold subject to the general terms and conditions of commercial sale, as published at https://www.nxp.com/profile/terms, unless otherwise agreed in a valid written individual agreement. In case an individual agreement is concluded only the terms and conditions of the respective agreement shall apply. NXP Semiconductors hereby expressly objects to applying the customer's general terms and conditions with regard to the purchase of NXP Semiconductors products by customer.

Export control — This document as well as the item(s) described herein may be subject to export control regulations. Export might require a prior authorization from competent authorities.

All information provided in this document is subject to legal disclaimers.

© 2026 NXP B.V. All rights reserved.

Suitability for use in non-automotive qualified products — Unless this document expressly states that this specific NXP Semiconductors product is automotive qualified, the product is not suitable for automotive use. It is neither qualified nor tested in accordance with automotive testing or application requirements. NXP Semiconductors accepts no liability for inclusion and/or use of non-automotive qualified products in automotive equipment or applications.

In the event that customer uses the product for design-in and use in automotive applications to automotive specifications and standards, customer (a) shall use the product without NXP Semiconductors' warranty of the product for such automotive applications, use and specifications, and (b) whenever customer uses the product for automotive applications beyond NXP Semiconductors' specifications such use shall be solely at customer's own risk, and (c) customer fully indemnifies NXP Semiconductors for any liability, damages or failed product claims resulting from customer design and use of the product for automotive applications beyond NXP Semiconductors' standard warranty and NXP Semiconductors' product specifications.

HTML publications — An HTML version, if available, of this document is provided as a courtesy. Definitive information is contained in the applicable document in PDF format. If there is a discrepancy between the HTML document and the PDF document, the PDF document has priority.

Translations — A non-English (translated) version of a document, including the legal information in that document, is for reference only. The English version shall prevail in case of any discrepancy between the translated and English versions.

Security — Customer understands that all NXP products may be subject to unidentified vulnerabilities or may support established security standards or specifications with known limitations. Customer is responsible for the design and operation of its applications and products throughout their lifecycles to reduce the effect of these vulnerabilities on customer's applications and products. Customer's responsibility also extends to other open and/or proprietary technologies supported by NXP products for use in customer's applications. NXP accepts no liability for any vulnerability. Customer should regularly check security updates from NXP and follow up appropriately.

Customer shall select products with security features that best meet rules, regulations, and standards of the intended application and make the ultimate design decisions regarding its products and is solely responsible for compliance with all legal, regulatory, and security related requirements concerning its products, regardless of any information or support that may be provided by NXP.

NXP has a Product Security Incident Response Team (PSIRT) (reachable at PSIRT@nxp.com) that manages the investigation, reporting, and solution release to security vulnerabilities of NXP products.

NXP B.V. — NXP B.V. is not an operating company and it does not distribute or sell products.

Trademarks

Notice: All referenced brands, product names, service names, and trademarks are the property of their respective owners.

NXP — wordmark and logo are trademarks of NXP B.V.

© 2026 NXP B.V. All rights reserved.

Please be aware that important notices concerning this document and the product(s) described herein, have been included in section 'Legal information'.

© 2026 NXP B.V. All rights reserved.

For more information, please visit: https://www.nxp.com

Date of release: 29 July 2026
Document identifier: MCXCP048M072F20RM