Freescale Semiconductor
Application Note

Document Number: AN4175
Rev. 1, 12/2012

MPC5674F to MPC5676R Migration Guide

by: David Erazmus

Contents

1 Introduction

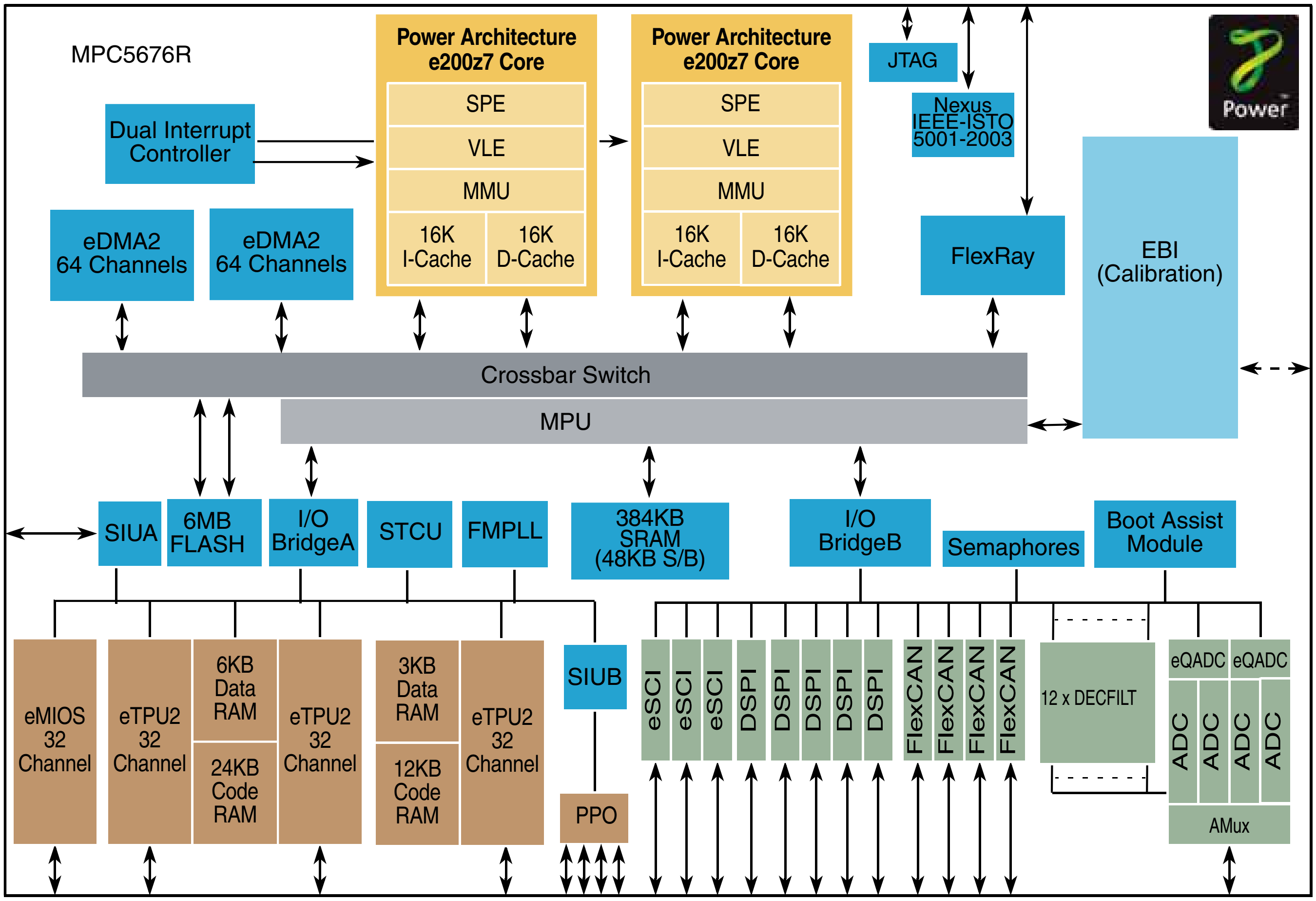
The MPC5676R is a high-performance 32-bit microcontroller built on Power Architecture™ technology targeted for powertrain applications. The two e200z7 host processor cores of the MPC5676R are compatible with the Power Architecture Book E architecture. They are user-mode compatible (except for floating point) with the classic PowerPC instruction set. The cores use Embedded Floating Point (EFP) rather than the Classic PowerPC floating-point instructions. The Book E architecture has enhancements that improve the architecture's fit in embedded applications. In addition to the standard and VLE Power Architecture instruction sets, this core has additional instruction support for digital signal processing (DSP).

The MPC5676R has two levels of memory hierarchy: separate 16 KB instruction and 16 KB data caches for each of the two cores and 384 KB of on-chip SRAM. There is 6 MB of internal flash. An external bus interface is also available for special packaged parts to support application development and calibration.

2 Overview

The MPC5676R is a pin-compatible replacement and upgrade for the MPC5674F. This document provides a summary of the differences between the MPC5674F and MPC5676R devices and may be used as a guide for planning a migration to the MPC5676R. (The information provided in this application note also applies to any other device in the MPC567xR family.)

Though the MPC5676R can be used as a drop-in replacement for the MPC5674F in an existing design, some differences in the operation of the device must be taken into account. Depending on your application, software modifications may be necessary. This document covers both the differences and the added features and functionality of the MPC5676R.



LEGEND

ADC	– Analog to Digital Convertor
AMux	– Analog Pin Multiplexer
CRC	– Cyclic Redundancy Check
D-Cache	– Data Cache
DECFILT	– Decimation Filter
DSPI	– Deserial/Serial Peripheral Interface
DTS	– Development Tool Semaphore
EBI	– External Bus Interface
ECSM	– Error Correction Status Module
eDMA2	– Enhanced Direct Memory Access controller version 2
eMIOS	– Enhanced Modular I/O System
eQADC	– Enhanced Queued Analog to Digital Converter
eSCI	– Enhanced Serial Communications Interface
eTPU2	– Enhanced Time Processing Unit version 2
FlexCAN	– Flexible Controller Area Network controller
FMPLL	– Frequency Modulated Phase Lock Loop clock generator
I-Cache	– Instruction Cache
IRC	– Internal RC Oscillator
JTAG	– Joint Test Action Group controller
MMU	– Memory Management Unit
MPU	– Memory Protection Unit
PIT	– Periodic Interval Timer
PMU	– Power Management Unit
PPO	– Protected Port Output
S/B	– Stand-by
SIUA	– System Integration Unit A
SIUB	– System Integration Unit B
SPE	– Signal Processing Engine
SRAM	– Static RAM
STCU	– Self Test Control Unit
STM	– System Timer Module
SWT	– Software Watchdog Timer
VLE	– Variable Length instruction Encoding

Figure 1. MPC5676R block diagram

NOTE

For clarity, the following modules are omitted from the above diagram: PMU, SWT, STM, PIT, ECSM, DTS, and CRC.

The following table is a summary of the feature differences between the MPC5674F and MPC5676R devices.

Table 1. MPC5674F/MPC5676R feature comparison

Feature	MPC5674F	MPC5676R
Core	e200z760	e200z759
Number of cores	1	2
MMU entries	64	32 per core
Software Watchdog Timer (SWT)	1	2
SRAM	256 KB (32 KB Standby)	384 KB (48 KB Standby)
Flash memory	4 MB	6 MB
DMA channels	64 + 32	64 + 64
SPI	4	5
SPI Parity Generation / Error Checking	No	Yes
eTPU channels	64	96
eTPU version	2 × eTPU2	3 × eTPU2
eTPU code memory	24 KB	24 KB + 12 KB
eTPU data memory	6 KB	6 KB + 3 KB
Number of interrupt vectors	448	467
Decimation filters	8	12
Protected port output	No	Yes
Self test controller	No	Yes
Development trigger semaphores	None	32
Cache Coherency Module	No	Yes
eQADC streaming	eQADC_B only	eQADC_A and eQADC_B
CRC modules	None	3
MMU external control	No	Yes
Internal RC oscillator	No	Yes
Extended voltage range PMC	No	Yes
EBI enhancements	No	Yes
Modules with Error Correcting Code (ECC)	None	Flexray, eTPU2

3 Memory

This section provides details of the differences in the embedded memories, including information that is relevant for porting the application to the MPC5676R.

3.1 SRAM

The MPC5676R has 384 KB of SRAM, compared to the 256 KB of SRAM on the MPC5674F. However, the Boot Assist Module (BAM) MMU initialization has not been altered. It will still allocate 256 KB of address space to the SRAM block at boot time for either flash or serial boot mode. If the extra SRAM is to be used, the application initialization code must extend this range in the MMU entry and also initialize all SRAM up to 384 KB. Additionally, linker directive files must be updated to reflect the change in the memory map.

3.2 Flash memory

- The MPC5676R has 6 MB of flash memory, increased from the 4 MB on the MPC5674F.
- The MPC5676R uses the same flash memory programming algorithm.
- Any tools used for programming the flash memory will need to be updated to be aware of the larger flash area, but they should still function as they did for MPC5674F if accessing only the lower 4 MB of the array.
- The low and mid address spaces within the flash array are identical for MPC5676R and MPC5674F. The high address space has been extended to cover the 6 MB range.
- ECC has been improved to detect address errors and more multi-bit errors.
- The MPC5676R flash controller has two system crossbar slave ports instead of one. One port is accessible only by core 0 while all other masters, including core 1, may access the second port. Arbitration between these two ports by the flash controller can be configured either as round-robin or fixed-priority. In fixed-priority mode, one port is assigned a higher priority than the other port and takes precedence whenever simultaneous accesses occur. In round-robin mode, the controller alternates servicing each port in turn.
- Some changes have also been made in the MPC5676R flash controller programming model to accommodate the dual port design. Specifically some control fields in the Flash BIUCR register have been moved to a new register, BIUCR3. Refer to the *MPC5676R Microcontroller Reference Manual* for details.

Table 2. MPC5674F vs. MPC5676R flash memory partitions

MPC5674F				MPC5676R			
Array A		Array B		Array A		Array B	
Block size	Use	Block size	Use	Block size	Use	Block size	Use
8 × 16 KB 2 × 64 KB	Low address space	—	—	8 × 16 KB 2 × 64 KB	Low address space	—	—
2 × 128 KB	Mid address space	—	—	2 × 128 KB	Mid address space	—	—
—	—	256 KB	Low address space	—	—	256 KB	Low address space
—	—	256 KB	Mid address space	—	—	256 KB	Mid address space
6 × 256 KB	High address space	6 × 256 KB	High address space	6 × 256 KB	High address space	6 × 256 KB	High address space
—	—	—	—	4 × 256 KB	High address space	4 × 256 KB	High address space

4 Core

The MPC5676R employs two e200z759 cores, instead of the single e200z760 core used on the MPC5674F. The e200z759 variant is based on the e200z760 with the following feature changes:

- The number of MMU entries has been reduced from 64 to 32 for each of the MPC5676R's two cores. Thus the device retains the same number of entries as MPC5674F but divides them between the cores.
- Each of the two cores operates at up to 184 MHz vs. the single core of the MPC5674F which operates at up to 264 MHz.
- MPC5676R uses Signal Processing Extension (SPE) version 1.1 instead of the SPE version 2 supported by MPC5674F.
- MPC5674F includes a Parallel Signature Unit (PSU) feature that can be used to test logic in the core by executing a specific set of instructions designed to provide coverage. MPC5676R adds another self-test feature: The ability to

perform a user-initiated Logic Built-In Self Test (LBIST) on the cores. Once initiated, LBIST runs without application intervention but does destroy the logic state of the core it is run on. Due to the length of the test (~15 ms) it is not automatically executed at reset and must be initiated by application control. If time is available to run the test at startup, user initialization code can begin the test and reset the core afterward. Alternatively, the LBIST can be run at a later time by having one core execute LBIST on the other core, and vice versa. The rate at which scan-chain information is clocked into the core during LBIST can be altered to reduce power consumption of the test.
- The instruction cache now features ECC rather than just a parity bit.
- The Cache Coherency Unit (CCU) in MPC5676R supports cache coherency between the two cores, including writes via the Nexus read-write access block and DMA writes to system memory.
- The size of the Nexus3 message buffer on each of MPC5676R's two cores is four times that of the MPC5674F core.

5 Pin function changes

The MPC5676R is pin-compatible with the MPC5674F but adds the following new functionality:

- DSPIE signals added as alternate pin functions.
- Added GPIO functionality on some Nexus trace pins.
- eTPU_C functions activated (these were reserved for future use in MPC5674F's pin map).
- Removed eQADC Synchronous Serial Interface (SSI) pin functions. These pin functions were unavailable and unsupported in MPC5674F but have now officially been removed from the pin map.

The following table summarizes these changes. Additions are shown in **bold**. Deletions are shown in (parentheses).

Table 3. Pin function changes

GPIO function	Primary function 1	Alternate function 1
GPIO445	ETPUC4	**PCSE1**
GPIO446	ETPUC5	**PCSE2**
GPIO447	ETPUC6	**PCSE3**
GPIO448	ETPUC7	**PCSE4**
GPIO449	ETPUC8	**PCSE5**
GPIO97	PCSA1	**PCSE0**
GPIO98	PCSA2	**SOUTE**
GPIO99	PCSA3	**SINE**
GPIO100	PCSA4	**SCKE**
GPIO240	PCSC2	(SDS)[1]
GPIO241	PCSC3	(SDO)[1]
GPIO242	PCSC4	(SDI)[1]
GPIO243	PCSC5	(FCK)[1]
GPIO220	MDO0	—
GPIO221	MDO1	—
GPIO222	MDO2	—
GPIO223	MDO3	—

[1] Function removed

6 Reset and clocks

There are several changes and improvements in the reset behavior and clock architecture of the MPC5676R:

- MPC5676R boots from a 16 MHz internal RC oscillator (IRC).
- Search order for the boot address in flash memory has changed.
- Full speed (1:1) clock mode for CPU and peripherals is no longer available.
- $\overline{\text{RSTOUT}}$ duration is different for some reset sources.
- MPC5676R adds memory-mapped reset vector registers for each core.

6.1 Internal RC Oscillator

Unlike the MPC5674F—which waits for the PLL to lock before exiting reset and starting with the system clock switched to the PLL—the MPC5676R boots from an Internal RC Oscillator (IRC) running at approximately 16 MHz. The PLL is not the source of the system clock until user application code selects it. If the Boot Assist Module (BAM) finds no user application code, it switches to Serial Boot Mode and automatically selects the PLL as clock source in order to provide a more accurate baud rate for the serial download. This provides a faster startup time for the system. The default clock input to the Software Watchdog Timers (SWT) has also been changed to the IRC instead of the crystal/external oscillator input (XTAL) in order to improve the ability to detect failures of the clock system.

6.2 Flash memory boot search order

The order in which the BAM searches for a valid Reset Configuration Half-Word (RCHW) has been changed for MPC5676R. The first address checked for the RCHW is 0x0002_0000 instead of 0x0000_0000 as on the MPC5674F.

Table 4. Flash memory boot search order

Block	Address on MPC567xF	Address on MPC567xR
0	0x0000_0000	0x0002_0000
1	0x0000_4000	0x0003_0000
2	0x0001_0000	0x0000_0000
3	0x0001_C000	0x0000_4000
4	0x0002_0000	0x0001_0000
5	0x0003_0000	0x0001_C000

6.3 Clock ratios

MPC5676R does not provide the 'legacy" clock ratio available in MPC5674F. In that mode, the CPU, eTPU, platform, and peripheral clocks could all run at the same speed (max 132 MHz). On the MPC5676R, platform and peripheral clocks are always 1/2 of the CPU clock while the option to clock the eTPU at the same frequency ("full" mode) or 1/2 the CPU frequency ("enhanced" mode) remains. Note, however, that the maximum clock speeds in these modes differ between the two devices. See the table below.

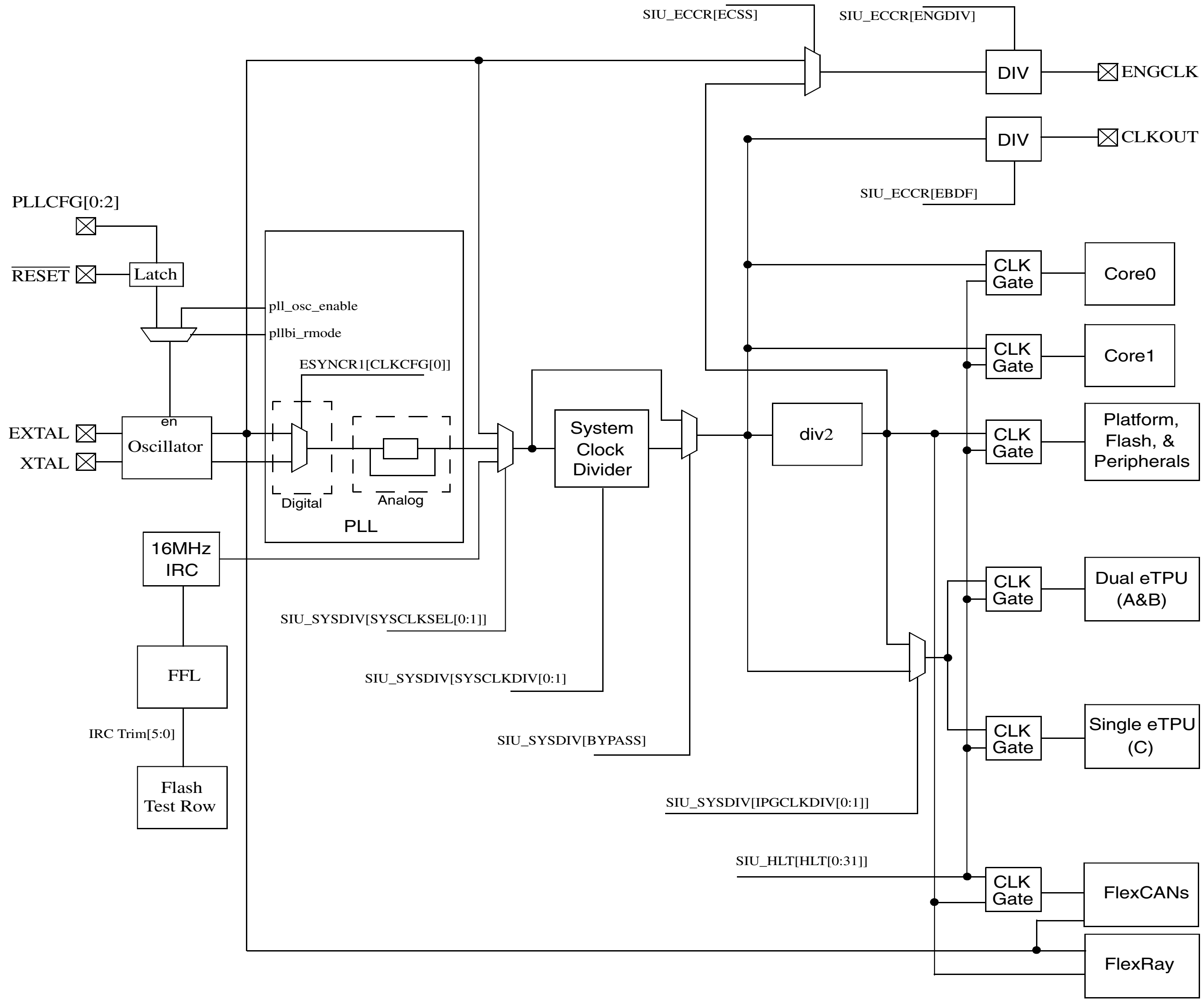


Figure 2. System level clock diagram

Table 5. System clock modes

SIU_SYSDIV [IPCLKDIV]	Mode	Description	MPC5674F max clock rate	MPC5676R max clock rate
0b00	Enhanced	CPU frequency is doubled. Platform,peripheral, and eTPU clocks are 1/2 of CPU frequency	264 MHz	200 MHz[1]
0b01	Full	CPU and eTPU frequency is doubled. Platform and peripheral clocks are 1/2 of CPU frequency	200 MHz	200 MHz[1]
0b10	—	Reserved	—	—
0b11	Legacy	CPU, eTPU, platform, and peripheral’s clocks all run at the same speed	132 MHz	Not supported

[1] MPC5676R is committed to meet 180 MHz plus FM frequency. 200 MHz is a design target.

6.4 Reset and clocks

When a reset occurs, in some cases, the duration that the $\overline{\text{RSTOUT}}$ signal is driven differs between MPC5674F and MPC5676R. There are also additional reset sources in MPC5676R.

Table 6. $\overline{\text{RSTOUT}}$ duration (in system clock cycles) by source

Reset source	MPC5674F duration[1]	MPC5676R duration
Power on reset	2400	2400
External reset (RESET pin)	2410	2310
Loss-of-lock	2420	2420
Loss-of-clock	2460	2430
Watchdog Timer or Debug reset from core 0	2430	2440
Software Watchdog Timer 0	2450	2460
Software Watchdog Timer 1	N/A	2470
Watchdog Timer or Debug Reset from core 1	N/A	2480
Simultaneous core 0 and core 1 reset	N/A	2490
Self Test Control Unit	N/A	2500
Software system reset	2470	2510
Software external reset	2480	2520 (IRC) / 17200 (other)
Checkstop reset[2]	2460	N/A

[1] In some clock modes, the MPC5674F uses a longer count value for the reset duration. Refer to the *MPC5674F Microcontroller Reference Manual*.

[2] This reset source is not available on e200z7xx Power Architecture Cores, though it was documented in the *MPC5674F Microcontroller Reference Manual*.

6.5 Reset vector registers

The MPC5676R adds two registers in the System Integration Unit for controlling reset of the individual cores: SIU_RSTVEC0 and SIU_RSTVEC1. SIU_RSTVEC0 shows the reset vector, reset control, and instruction execution type selection for Core 0. SIU_RSTVEC1 shows the equivalent details for core 1. The RST bit in each register is qualified by the corresponding core halt bit in the SIU halt register, such that both must be asserted for a core reset to occur. The order in which the RST and core halt bits is asserted does not matter. It is recommended that the RST bit is set immediately after the corresponding halt bit is set, to minimize the opportunity for a “single upset event” to prematurely reset a core.

After any reset, core 0's halt and RST bits are negated to allow core 0 to execute while core 1's halt and RST bits remain asserted to prevent core 1 from executing code until it is initialized by core 0.

6.5.1 SIU_RSTVEC0 and SIU_RSTVEC1 registers

SIU_RSTVEC0 and SIU_RSTVEC1 contain the reset vector, reset control, and instruction set selection for core 0 and core 1, respectively.

Table 7. SIU_RSTVEC0 and SIU_RESTVEC1 (0xC3F9_09AC, 0xC3F9_09B0)

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

Table 8. SIU_RSTVEC0 and SIU_RSTVEC1 field descriptions

Field	Description
RSTVEC	Reset Vector. Determines the initial program counter upon exiting reset. On POR, the value contained in the register defaults to 0xFFFF_FFFC, so that the core fetches Book E code from the BAM starting at address 0xFFFF_FFFC. User code may change this value to select a different fetch address on exit from a user initiated reset of this core.
RST	Controls the assertion of RESET to the core. Writing 1 to this bit causes the core to enter reset, provided the core's halt bit is also set. Reads of this bit indicate whether the core is being held in reset. 0—Core not in reset 1—Core reset requested
VLE	VLE Select. Selects whether the core executes VLE or Book E code at the reset vector address 0—Book E 1—VLE

7 Crossbar Switch (XBAR)

The system crossbar switch now includes additional master ports for the second core's instruction and data and an additional slave port for core 0 access to the flash memory. The following two tables show the added master and slave ports for the MPC5676R in bold.

Table 9. Crossbar master ports

Module	Port	Master ID
e200z7 core0—CPU instruction	Master 0	0

Table continues on the next page...

Table 9. Crossbar master ports (continued)

Module	Port	Master ID
e200z7 core0—data	Master 1	0
Nexus 3		8
e200z7 core1—CPU instruction	**Master 2**	**1**
e200z7 core1—data	**Master 3**	**1**
Nexus 3		**9**
eDMA_A	Master 4	4
eDMA_B	Master 5	5
FlexRay	Master 6	6
Reserved	Master 7	3

Unlike the MPC5674F, where the single flash memory port was accessible by all masters, the MPC5676R's two flash memory ports can be accessed by different masters as shown below.

Table 10. Crossbar slave ports

Module	Port	Accessible by Master Ports
On-chip flash memory (core 0 only)	Slave 0	0, 1
EBI (development bus)	Slave 1	0, 1, 2, 3, 4, 5, 6, 7
On-chip SRAM	Slave 2	0, 1, 2, 3, 4, 5, 6, 7
On chip flash memory (excluding core 0)	Slave 3	2, 3, 4, 5, 6, 7
Peripheral bridge A (PBRIDGE_A)	Slave 6	0, 1, 2, 3, 4, 5, 6, 7
Peripheral bridge B (PBRIDGE_B)	Slave 7	0, 1, 2, 3, 4, 5, 6, 7

Following is a diagram showing the crossbar connections on MPC5676R. Up to six paths can be in use at the same time; five example concurrent paths are highlighted in the figure. The highlighted paths show:

- e200z7 core 0 instruction fetch from the internal flash memory
- e200z7 core 1 instruction fetch from the internal flash memory
- e200z7 core 1 load or store from a peripheral on peripheral bridge B
- eDMA A access to or from a peripheral on peripheral bridge A
- FlexRay store to the internal SRAM

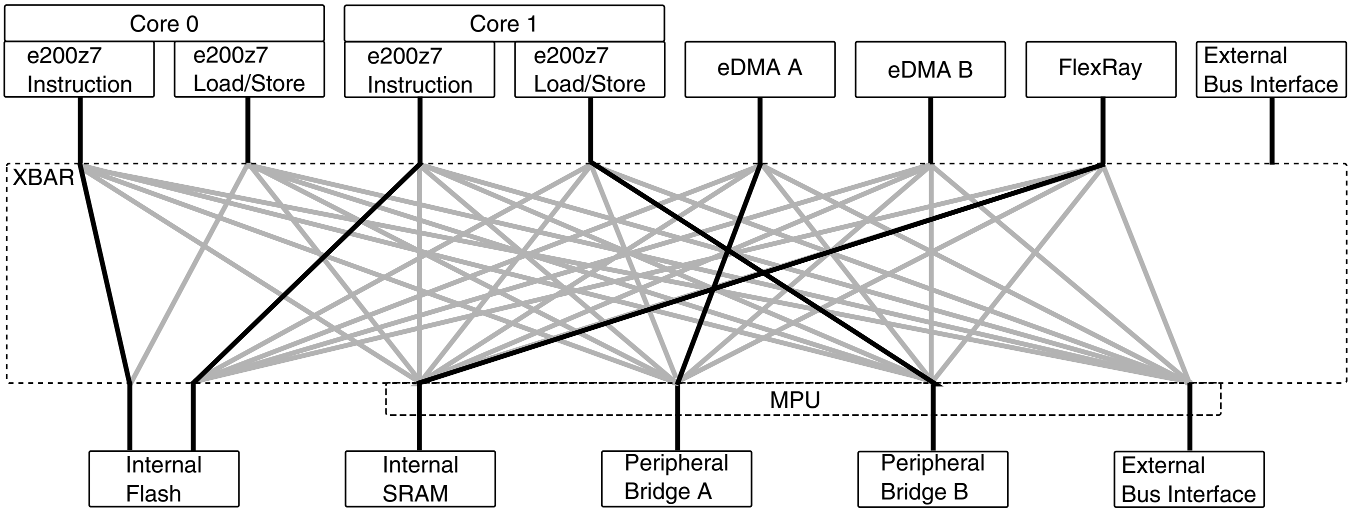


Figure 3. MPC5676R system crossbar

The default master priorities have also changed due to the new ports. The following table shows the differences.

Table 11. Master port default priorities

Master	MPC5674F priority	MPC5676R priority
0	0	0
1	1	1
2	—	2
3	—	3
4	2	4
5	3	5
6	4	6
7	5	7

8 External Bus Interface (EBI)

The External Bus Interface (EBI) handles the transfer of information between the internal buses and the memories or peripherals in the external address space. The EBI includes a memory controller that generates interface signals to support a variety of external memories. The MPC5676R EBI is compatible with MPC5674F but adds a few new features:

- Short Burst Length (SBL) allows for a two-word external burst. Four beats on a 16-bit bus or two beats on a 32-bit bus.
- Late Read/Write Negation (LWRN) determines the timing of the RD/$\overline{\text{WR}}$ signal negation for a write transfer (except when SETA=1). When enabled, the RD/$\overline{\text{WR}}$ signal is negated on the same cycle as chip-select negation, rather than one-cycle earlier as is normally the case.
- Guarantee Chip Select Negation (GCSN) allows support for guaranteeing that the EBI will negate $\overline{\text{CS}}$ between all back-to-back transfer cases (even those that are part of a set of small accesses). When enabled, this adds an extra dead cycle for some back-to-back cases.
- Early Output Enable (EOE) determines the timing of $\overline{\text{OE}}$ signal assertion for a read transfer. There are several timing options:
  - Assertion of $\overline{\text{OE}}$ one CLKOUT cycle after TS
  - Assertion of $\overline{\text{OE}}$ with TS assertion
  - Assertion of $\overline{\text{OE}}$ one internal clock cycle after TS

9 Peripherals

The following sections detail the differences and additions among the peripheral modules of MPC5676R.

9.1 Dual Interrupt Controller (INTC)

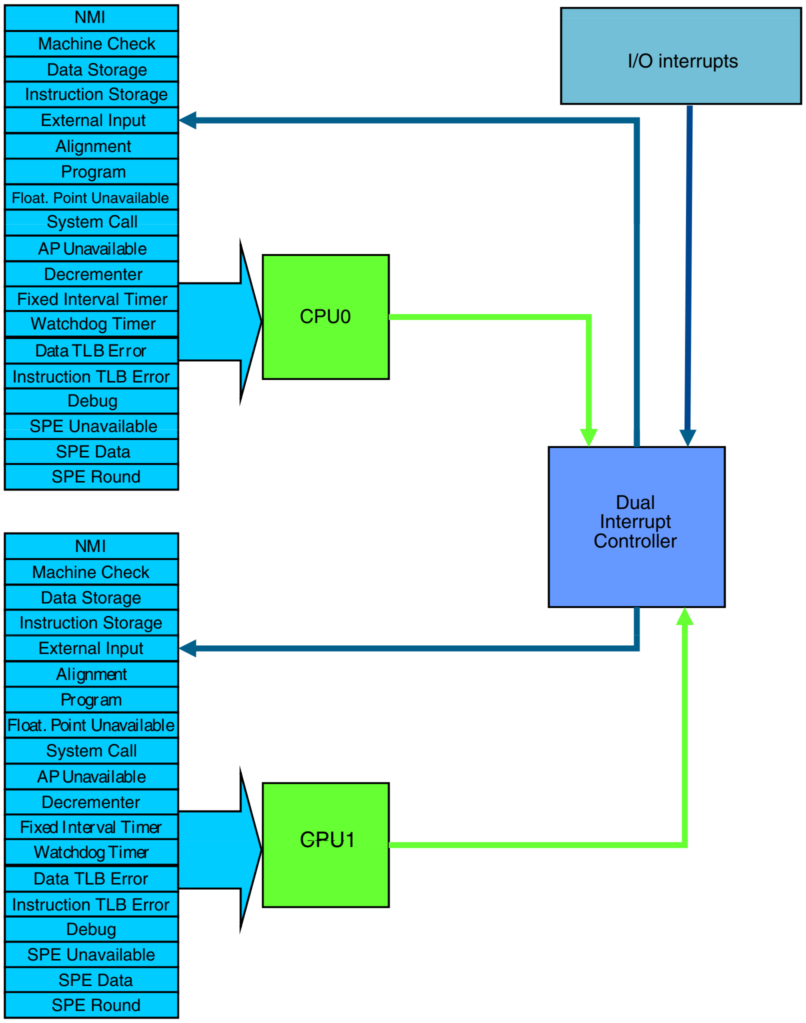


Figure 4. Dual Interrupt Controller

MPC5676R features a Dual Interrupt Controller supporting both cores.

- Each core has 19 interrupt vectors to handle:
  - User programmed interrupts
  - External interrupt from Dual Interrupt Controller
  - Non-Maskable Interrupt (NMI)
  - Interrupts caused by software or hardware faults
- Over 448 interrupt sources from multiple on-chip resources including FMPLL, DSPI, FlexCAN, Flexray, eTPU, eMIOS, and external pins
- I/O interrupts can be steered to either core or both cores at the same time
- Also supports software interrupt Fixed Interval Timer signaling between cores or to the same core

- Two operating modes:
    - Software Vector mode, which optimizes code size
    - Hardware Vector Mode, which optimizes response time
- Dual Interrupt Controller hardware arbitration reduces I/O interrupt latency

9.2 Peripheral Bridges (PBRIDGE)

The MPC5676R, like the MPC5674F, has two Peripheral Bridges (PBRIDGE_A and PBRIDGE_B). These interface between the system bus and lower speed peripherals. The version of PBRIDGE used on MPC5676R does not have configurable per-module write buffering or per-master access protections. It does, however, add the ability to terminate a transaction with a bus error if an access to a peripheral is attempted when the peripheral's clock is disabled.

9.3 Enhanced Queued Analog Digital Converters (eQADC)

The eQADC and decimation filter modules in MPC5676R are fully backward-compatible for existing MPC567xF applications. MPC5676R adds new features and more flexibility in using these modules.

In addition to providing more decimation filters, MPC5676R introduces an alternate method for selecting filters as destinations for eQADC conversion results. As well, the decimation filters can be accessed from either eQADC module where previously, in the MPC5674F, they could only be accessed from eQADC_B. This allows more flexibility in routing to one or two filter destinations. The MPC5674F method of addressing filters is still supported for compatibility with existing software.

Decimation Filter Triggered Output mode is an alternative for controlling decimation of filter output. A selection of eTPU channels can gate filter output on/off rather than using the decimation counter register.

eQADC result timestamps may now be imported from eTPU modules via the Shared Time and Angle Count (STAC) bus. Streaming mode conversions are now supported for both eQADC modules, whereas on the MPC5674F streaming mode conversions were only available on eQADC_B.

MPC5676R adds the ability to select odd clock prescaler values via the ODD_PS field in the ADC control registers. A field, CLK_DTY, allows for the duty cycle of the prescaled clock to be greater or less than 50% when there are an odd number of peripheral clocks in the resulting ADC clock period.

The following table summarizes the differences between MPC5674F and MPC5676R for the eQADC modules.

Table 12. eQADC and decimation filter differences

eQADC features	MPC5674F	MPC5676R
Alternate configuration registers	8	14
Enhanced decimation filters	8	12
Enhanced decimation filter selection	—	Yes
Decimation Filter Triggered output mode	—	Yes
Decimation filter access	eQADC_B only	eQADC_A and eQADC_B
Import timestamp from timer via STAC bus	—	Yes
Enhanced queue triggers from eTPUC	—	Yes
Streaming mode	eQADC_A only	eQADC_A and eQADC_B
Clock prescaler odd rates selection	—	Yes

9.4 Enhanced Serial Communications Interface (eSCI)

MPC5676R uses a newer version of the eSCI module than the MPC5674F. Full details of the eSCI for MP5676R can be found in *MPC5676R Microcontroller Reference Manual*. The following is a summary of the differences.

Table 13. MPC5676R Enhanced Serial Communications Interface (eSCI)

Register	Change
Control Register 1 (eSCI_CR1)	Removed the ILT field. Idle line detection always starts after last stop bit reception. Equivalent to ILT = 1 setting in MPC5674F. Idle line detection after data bit reception is removed.
Control Register 3 (eSCI_CR3)	Removed the SYNM field. Re-sync is performed at falling start bit edge only. Equivalent to SYNM = 1 setting in MPC5674F. Performing re-sync at start of data bit edge has been removed.
Interrupt Flag and Status Register 1 (eSCI_IFSR1)	Added the following fields: • LACT: LIN active status bit. Indicates an ongoing LIN transaction. • WACT: LIN wake up active status bit. Indicates an ongoing LIN wake up signal reception. • DACT: DMA Active. This bit is set when a transmit or receive DMA request is pending.
LIN Transmit Register (eSCI_LTR)	The last data written into this register can be read out.
Interrupt Flag and Status Registers 1 and 2 (eSCI_IFSR1/2)	• OR and PF: flags are set after 10th sample of the last stop bit. After stop bit verification. (For the MPC5674F, flags are set after 16th sample of the last stop bit. After end of stop bit.) • RDRF: flag is set after 10th sample of the last stop bit. After stop bit verification. Flag is not set in DMA RX mode. (For the MPC5674F, the flag is set after 16th sample of the last stop bit. After end of stop bit. Flag is set in DMA RX mode.) • UREQ, BERR, OVFL, CERR, CKERR: flags are set after 10th sample of the last stop bit. After stop bit verification. (For the MPC5674F, flags are set after 16th sample of the last stop bit. After end of stop bit.) • RXRDY: flag is set after 10th sample of the last stop bit. After stop bit verification. Flag is not set in DMA RX mode. (For the MPC5674F, the flag is set after 16th sample of the last stop bit. After end of stop bit. Flag is set in DMA RX mode.)

9.5 Deserial Serial Peripheral Interface (DSPI)

MPC5676R uses a newer version of the DSPI module than the MPC5674F. Full details of the DSPI for MP5676R can be found in the *MPC5676R Microcontroller Reference Manual*. The following is a summary of the differences.

- MPC5676R adds a fifth DSPI module, DSPI_E, which is identical to the others except that it supports serialization only via the DSPI_ADSR register.
- DSPI parity functionality and associated parity error status bits and interrupts are new in this version. The two new interrupts, signalled by SPEF and DPEF, are combined with overflow and underflow interrupt requests in each module.
- There is a new Deserialized Data Match interrupt combined with the Transfer Complete (TCF) interrupt request.
- Slave and DSI frames may now be up to 32 bits long.
- RXFIFO and DSI registers are now 32 bits wide.

9.6 Enhanced Time Processing Unit (eTPU2)

There are two major differences in the enhanced Timing Processor Unit 2 (eTPU2) implementation on the MPC5674F and the MPC5676R. There is a third eTPU2 on the MPC5676R, providing a total of 96 timer channels. The eTPU2 memories implement an Error Correction Code (ECC). The ECC allows correction of single bit errors and detection of double bit errors.

Each eTPU on both the MPC5674F and the MPC5676R implement 32 channels. There are two eTPU2s on the MPC5674F, but there are three eTPU2s on the MPC5676R.

The eTPU2 memory consists of the Shared Code Memory (SCM), used for eTPU2 program code, and the Shared Parameter RAM (SPRAM) used for passing parameters between the eTPU2 and the host processor (either of the e200z7 cores). The following table shows the configuration of the eTPUs.

Table 14. eTPU feature comparison

eTPU	MPC5674F			MPC5676R		
	SCM memory size	PSRAM memory size	Number of channels	SCM memory size	PSRAM memory size	Number of channels
eTPU_A	24 KB[1]	6 KB[2]	32	24 KB[1]	6 KB[2]	32
eTPU_B			32			32
eTPU_C	—	—	—	12 KB	3 KB	32

1. eTPU_A and eTPU_B share the same Shared Code Memory (SCM).
2. eTPU_A and eTPU_B share the Shared Parameter RAM (SPRAM).

The ECC bits in both sets of SRAM should be initialized by the host processor prior to use. Loading the SCM with microcode will initialize the SCM RAM used by the eTPU2 code image; however, to support the Multiple Input Signature Calculator (MISC) function of the eTPU2, the entire SCM memory space should be initialized to a known value. The ECC on the SCM is based on 32-bit words, so 32-bit memory writes should be used by the host processor to initialize the SCM.

All of the SPRAM should also be initialized by the host processor during the system initialization. The ECC on the SPRAM is based on 8-bit words.

Table 15. eTPU memory map differences

eTPU	Memory use	MPC5674F		MPC5676R	
		Start address	End address	Start address	End address
eTPU_A and eTPU_B	Registers	0xC3FC_0000	0xC3FC_0BFF	0xC3FC_0000	0xC3FC_0BFF
	SPRAM (6 KB)	0xC3FC_8000	0xC3FC_97FF	0xC3FC_8000	0xC3FC_97FF
	SPRAM PSE[1] mirror (6 KB)	0xC3FC_C000	0xC3FC_D7FF	0xC3FC_C000	0xC3FC_D7FF
	SCM (24 KB)	0xC3FD_0000	0xC3FD_5FFF	0xC3FD_0000	0xC3FD_5FFF
eTPU_C	Registers	—	—	0xC3E2_0000	0xC3E2_0BFF
	SPRAM (3 KB)	—	—	0xC3E2_8000	0xC3E2_8BFF
	SPRAM PSE[1] mirror (3KB)	—	—	0xC3E2_C000	0xC3E2_CBFF
	SCM (12KB)	—	—	0xC3E3_0000	0xC3E3_2FFF

1. Parameter Sign Extension access area

NOTE

ECC registers have been added to previously reserved space in each "Registers" block shown in the above table at offsets 0x100 through 0x13F.

In addition to the above major changes, there has also been a slight timing change to the Time Slot Transition (TST) between two consecutive threads running on the same channel. The TST has been modified so that when switching between different channels, a match is now disabled during the first two cycles of the TST. On the MPC5674F's eTPU2, a match was always automatically enabled during the TST, even if it was disabled in the threads. To ensure the MPC5676R's eTPU2 has the same behavior as the previous eTPU2, the TST has been extended by one cycle to enable a match and thus mimic the behavior of the previous eTPU2. This condition only applies to the case that the TST is between threads of the same channel. When switching between different channels a match is always enabled for the channel that is not being serviced.

9.7 Flexray

The Flexray module in MPC5676R is functionally backward-compatible with that in the MPC5674F with the following exceptions:

- The Message Buffer Configuration, Status, and Control registers have been moved from offset 0x100 in the Flexray memory map to offset 0x800.
- The Message Buffer Data Field Offset registers (FR_MBDORx) now appear in the memory map starting at offset 0x1000 rather than being located in main memory.
- The interrupt source vector numbers for Flexray are different in MPC5676R. Refer to the Interrupt Controller section of the *MPC5676R Microcontroller Reference Manual* for details.

In addition, the Flexray module now includes an Error Correcting Code (ECC) feature. The ECC control and status registers appear at offset 0x00F0 in the Flexray memory map. LRAM ECC test registers appear at offset 0x1108. Refer to the *MPC5676R Microcontroller Reference Manual* for full details.

9.8 Semaphores

To support dual-processor operation, MPC5676R provides two hardware solutions for implementing semaphores, a common technique for managing shared resources. The Power Architecture reservation mechanism, which uses special assembly language instructions to reserve and release specified memory locations for each core, is one approach. The other is a peripheral block accessible through memory-mapped reads/writes across the peripheral bus.

9.8.1 Reservations

In MPC5676R, the Power Architecture reservation mechanism present in MPC5674F has been extended to support both cores.

Reservations require the use of special assembly language instructions to load-and-reserve, then store conditionally. Rather than locking a memory location, this provides a way for a software routine to know when it has not successfully performed a read-modify-write. For example:

1. Load a memory word and obtain a reservation for that word.
2. Modify the value.
3. Store the value conditionally if the reservation is still held.

Step 3 above fails if some other thread on the same core or a different core modifies the reserved location. This functionality can be used to implement an atomic read-modify-write by re-trying the above steps until they succeed.

In previous devices such as the MPC5674F, this mechanism did not reserve a particular location but instead was considered as reserving the entire memory space. As such, it didn't inform that another thread had modified the desired location but that another thread had modified some location somewhere.

In the MPC5676R, this procedure has been enhanced to reserve the specific memory location given in the load-with-reservation instruction. Each of the two cores can maintain a separate reservation if desired. If they reserve the same location, only the first to store-conditionally with that reservation will succeed. The other will lose the reservation and fail the subsequent store.

In all cases, each core may only maintain one reservation at a time. Creating a new reservation replaces the existing one.

Since the reservation mechanism allows for the implementation of an atomic read-modify-write, it can be used to implement a semaphore flag that software would use to lock access to a shared resource. For example:

```
addi    r4,0,1      # r4 = 1
lwarx   r5,0,r3     # load and reserve (r3 contains address of semaphore)
cmpwi   r5,0        # Already in use if semaphore
bne     LockFail    # is not equal to 0
stwcx.  r4,0,r3     # try to store non-zero
bne     LockFail    # If reservation was still present, lock was successful

# Enter critical section
...
# Exit critical section

addi    r4,0,0      # Unlock the semaphore
stw     r4,0(r3)
b       Done
LockFail:           # Other bus master has the lock so try again later
...                 # Lock failure code
Done:               # Lock done code
```

9.8.2 Hardware semaphore block

The semaphore block has been added to the MPC5676R to support dual-processor operations. Applications can use the hardware block as an alternative to (or in addition to) the core reservation feature.

The module supports 16 hardware-enforced gates (semaphores). Each gate is a three-state entity mapped as 2 bits with all 16 gates appearing in the memory map as an array of byte-addressable locations. Each gate's 2-bit state is defined as follows.

Table 16. Semaphore state

Value (binary)	State
00	Unlocked
01	e200z7 core 0
10	e200z7 core 1

Each core can obtain a lock to a specific gate by writing its value (as shown above) to the semaphore gate register. The core then reads back the gate register to determine if the lock succeeded (as shown in the following example).

```
this_core = processor_num() + 1; // Some assembly required to read processor number
gate[n] = this_core;       // Attempt lock by writing gate with processor_number + 1
if (gate[n] == this_core)  // read gate to verify lock was obtained
{                          // Semaphore lock obtained
                           // Enter critical section
                           // ...
                           // Exit critical section
  gate[n] = 0;             // Unlock the semaphore
}
else
{
                           // Other bus master has the lock so try again later
}
```

9.9 Protected port outputs

MPC5676R provides four GPIO outputs that can be restricted for access by one CPU core.

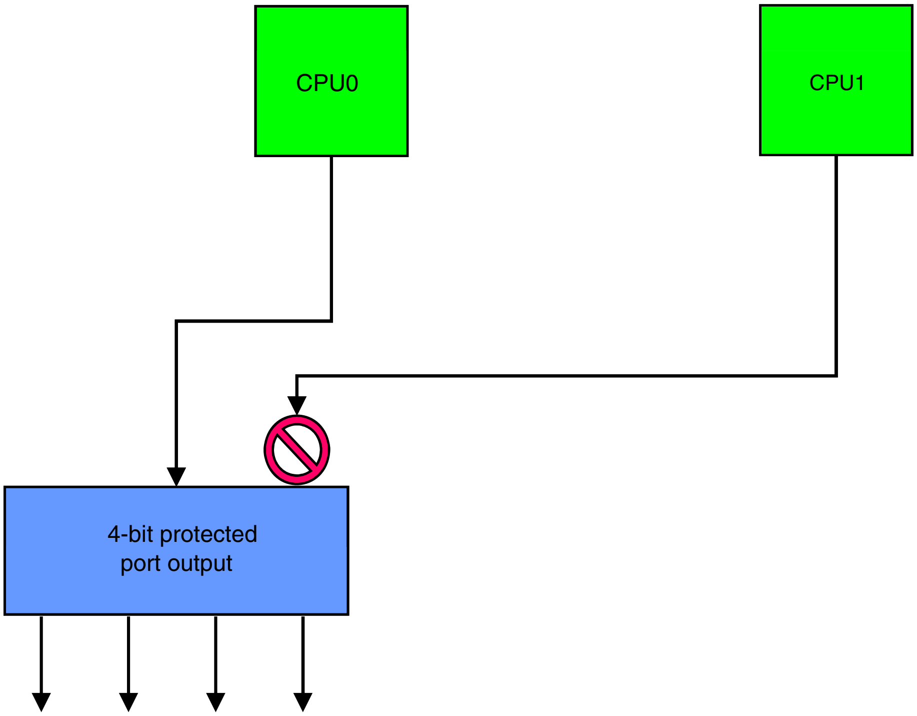


Figure 5. Example: protected port outputs restricted to core 0

By default, these output ports are accessible through SIU_A from either core. SIU_B contains a mirror of the control registers for these four pins. At reset application code running on core 0 can, before enabling core 1, restrict access by transferring control of these pins to the mirrored registers in SIU_B. Once this is done, any master access to the original control registers for these pins in SIU_A will have no effect on the pins.

9.10 Cyclic Redundancy Checker (CRC)

The CRC module is another addition to MPC5676R. It provides a fast on-chip capability for calculating CRC checksums. The CRC module supports three separate channels (or "contexts"), allowing three independent data streams to have checksums calculated at the same time.

The module supports two CRC algorithms. Either may be selected for each channel being calculated.

Equation 1: CRC-CCITT (x25 protocol)

$X^{16} + X^{12} + X^5 + 1$

Equation 2: CRC-32 (ethernet protocol)

$X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^8 + X^7 + X^5 + X^4 + X^2 + X + 1$

Data may be sent to the CRC module for calculation either via the CPU or DMA.

10 Debug features

MPC5676R provides some new and upgraded debug capabilities.

Table 17. Debug feature summary

Debug feature	MPC5674F	MPC5676R
Larger core message FIFO	—	X
Data address compare	2	2
Data trace start/end addresses	5	5
MMU manipulation via Nexus while the core is running (PID Control)	—	X
System clock available flag using MDO0	—	X
40 MHz JTAG	—	X
Trigger semaphores for data acquisition	—	32 raster DTS

10.1 Development Trigger Semaphore (DTS) overview

The Development Trigger Semaphore (DTS) module provides a 32-bit register of semaphores and an identification register. The identification register can be used as part of a triggered data acquisition protocol between an embedded controller and an external tool, either for calibration or for rapid prototyping.[1]

The DTS_SEMAPHORE register, along with the DTS Trigger Output ($\overline{\text{DTO}}$), provides a mechanism to indicate to the tool that the calibration variables (or sets of measurements), from 1 to 32, have been updated with new values and are available for access. Data coherency is maintained by limiting the setting of bits in the DTS_SEMAPHORE to only the CPU and DMA. The CPU and DMA cannot clear any bit in this register. Only a tool access via Nexus read/write access through the JTAG port can clear bits in this register. Access permissions to the DTS_SEMAPHORE register are based on the crossbar (XBAR) master identification number for the port.

The DTS_STARTUP register provides a mechanism for the tool to notify software running on the CPU that a tool is connected and can provide information about either the tool type or the options that can be used by the software.

The DTS_ENABLE register enables the DTS feature.

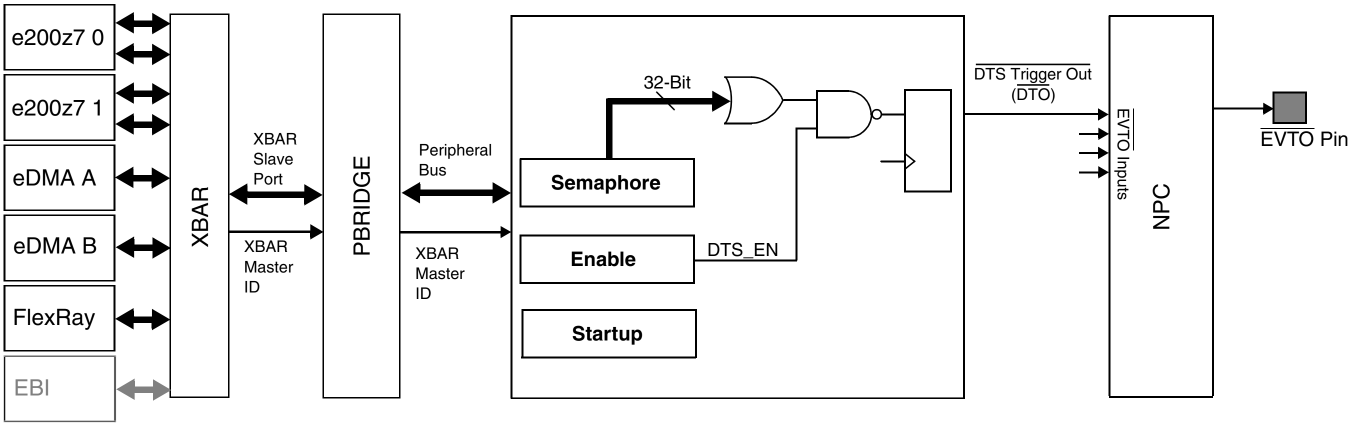


Figure 6. MPC567xR system

[1] This triggered data acquisition uses the Nexus read/write access via the JTAG interface of the Nexus debug port and is different than the data acquisition protocol that uses the Nexus auxiliary port and is defined in either the IEEE-ISTO 5001-2003 or IEEE-ISTO 5001-2010 Nexus standard. The IEEE-ISTO 5001 Nexus data acquisition is also supported on the e200z4 and e200z7 cores.

How to Reach Us:

Home Page:
www.freescale.com

Web Support:
http://www.freescale.com/support

USA/Europe or Locations Not Listed:
Freescale Semiconductor
Technical Information Center, EL516
2100 East Elliot Road
Tempe, Arizona 85284
+1-800-521-6274 or +1-480-768-2130
www.freescale.com/support

Europe, Middle East, and Africa:
Freescale Halbleiter Deutschland GmbH
Technical Information Center
Schatzbogen 7
81829 Muenchen, Germany
+44 1296 380 456 (English)
+46 8 52200080 (English)
+49 89 92103 559 (German)
+33 1 69 35 48 48 (French)
www.freescale.com/support

Japan:
Freescale Semiconductor Japan Ltd.
Headquarters
ARCO Tower 15F
1-8-1, Shimo-Meguro, Meguro-ku,
Tokyo 153-0064
Japan
0120 191014 or +81 3 5437 9125
support.japan@freescale.com

Asia/Pacific:
Freescale Semiconductor China Ltd.
Exchange Building 23F
No. 118 Jianguo Road
Chaoyang District
Beijing 100022
China
+86 10 5879 8000
support.asia@freescale.com

Information in this document is provided solely to enable system and software implementers to use Freescale Semiconductors products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits or integrated circuits based on the information in this document.

Freescale Semiconductor reserves the right to make changes without further notice to any products herein. Freescale Semiconductor makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale Semiconductor assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any liability, including without limitation consequential or incidental damages. "Typical" parameters that may be provided in Freescale Semiconductor data sheets and/or specifications can and do vary in different applications and actual performance may vary over time. All operating parameters, including "Typicals", must be validated for each customer application by customer's technical experts. Freescale Semiconductor does not convey any license under its patent rights nor the rights of others. Freescale Semiconductor products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which failure of the Freescale Semiconductor product could create a situation where personal injury or death may occur. Should Buyer purchase or use Freescale Semiconductor products for any such unintended or unauthorized application, Buyer shall indemnify Freescale Semiconductor and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claims alleges that Freescale Semiconductor was negligent regarding the design or manufacture of the part.

RoHS-compliant and/or Pb-free versions of Freescale products have the functionality and electrical characteristics as their non-RoHS-complaint and/or non-Pb-free counterparts. For further information, see http://www.freescale.com or contact your Freescale sales representative.

For information on Freescale's Environmental Products program, go to http://www.freescale.com/epp.