DSP56364 24-Bit Digital Signal Processor

Users Manual

Document Number: DSP56364UM
Rev. 2
08/2006

freescale semiconductor

How to Reach Us:

Home Page:
www.freescale.com

E-mail:
support@freescale.com

USA/Europe or Locations Not Listed:
Freescale Semiconductor
Technical Information Center, CH370
1300 N. Alma School Road
Chandler, Arizona 85224
+1-800-521-6274 or +1-480-768-2130
support@freescale.com

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)
support@freescale.com

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 Hong Kong Ltd.
Technical Information Center
2 Dai King Street
Tai Po Industrial Estate
Tai Po, N.T., Hong Kong
+800 2666 8080
support.asia@freescale.com

***For Literature Requests Only*:**
Freescale Semiconductor Literature Distribution Center
P.O. Box 5405
Denver, Colorado 80217
1-800-521-6274 or 303-675-2140
Fax: 303-675-2150
LDCForFreescaleSemiconductor@hibbertgroup.com

Contents

3 Memory Configuration

4 Core Configuration

5 General Purpose Input/Output Port (GPIO)

6 Enhanced Serial AUDIO Interface (ESAI)

7 Serial Host Interface

List of Figures

List of Tables

Preface

This manual contains the following sections and appendices.

SECTION 1—DSP56364 OVERVIEW

- Provides a brief description of the DSP56364, including a features list and block diagram. Lists related documentation needed to use this chip and describes the organization of this manual.

SECTION 2—SIGNAL/CONNECTION DESCRIPTIONS

- Describes the signals on the DSP56364 pins and how these signals are grouped into interfaces.

SECTION 3—MEMORY CONFIGURATION

- Describes the DSP56364 memory spaces, RAM and ROM configuration, memory configurations and their bit settings, and memory maps.

SECTION 4—CORE CONFIGURATION

- Describes the registers used to configure the DSP56300 core when programming the DSP56364, in particular the interrupt vector locations and the operation of the interrupt priority registers. Explains the operating modes and how they affect the processor's program and data memories.

SECTION 5—GENERAL PURPOSE INPUT/OUTPUT (GPIO)

- Describes the DSP56364 GPIO capability and the programming model for the GPIO signals (operation, registers, and control).

SECTION 6—ENHANCED SERIAL AUDIO INTERFACE (ESAI)

- Describes the full-duplex serial port for serial communication with a variety of serial devices.

SECTION 7—SERIAL HOST INTERFACE (SHI)

- Describes the serial input/output interface providing a path for communication and program/coefficient data transfers between the DSP and an external host processor. The SHI can also communicate with other serial peripheral devices.

APPENDIX A—BOOTSTRAP PROGRAM

- Lists the bootstrap code used for the DSP56364.

APPENDIX B—BSDL LISTING

- Provides the BSDL listing for the DSP56364.

APPENDIX C—PROGRAMMING REFERENCE

- Lists peripheral addresses, interrupt addresses, and interrupt priorities for the DSP56364. Contains programming sheets listing the contents of the major DSP56364 registers for programmer reference.

Manual Conventions

The following conventions are used in this manual:

- Bits within registers are always listed from most significant bit (MSB) to least significant bit (LSB).
- When several related bits are discussed, they are referenced as AA[n:m], where n>m. For purposes of description, the bits are presented as if they are contiguous within a register. However, this is not always the case. Refer to the programming model diagrams or to the programmer's sheets to see the exact location of bits within a register.
- When a bit is described as "set", its value is 1. When a bit is described as "cleared", its value is 0.
- The word "assert" means that a high true (active high) signal is pulled high to V_{CC} or that a low true (active low) signal is pulled low to ground. The word "deassert" means that a high true signal is pulled low to ground or that a low true signal is pulled high to V_{CC}.

High True/Low True Signal Conventions

Signal/Symbol	Logic State	Signal State	Voltage
$\overline{\text{PIN}}$[1]	True	Asserted	Ground[2]
PIN	False	Deasserted	V_{CC}[3]
PIN	True	Asserted	V_{CC}
PIN	False	Deasserted	Ground

1 PIN is a generic term for any pin on the chip.

2 Ground is an acceptable low voltage level. See the appropriate data sheet for the range of acceptable low voltage levels (typically a TTL logic low).

3 V_{CC} is an acceptable high voltage level. See the appropriate data sheet for the range of acceptable high voltage levels (typically a TTL logic high).

- Pins or signals that are asserted low (made active when pulled to ground)
  - In text, have an overbar (e.g., $\overline{\text{RESET}}$ is asserted low).
  - In code examples, have a tilde in front of their names. In example below, line 3 refers to the $\overline{\text{SS0}}$ pin (shown as `~SS0`).
- Sets of pins or signals are indicated by the first and last pins or signals in the set (e.g., HA1–HA8).
- Code examples are displayed in a monospaced font, as shown below:

Example Sample Code Listing

```
BFSET    #$0007,X:PCC; Configure:                    line 1

         ; MISO0, MOSI0, SCK0 for SPI master         line 2

              ; ~SS0 as PC3 for GPIO                 line 3
```

- Hex values are indicated with a dollar sign ($) preceding the hex value, as follows: $FFFFFF is the X memory address for the core interrupt priority register (IPR-C).
- The word "reset" is used in four different contexts in this manual:
  - the reset signal, written as "$\overline{\text{RESET}}$,"

— the reset instruction, written as “RESET,”
— the reset operating state, written as “Reset,” and
— the reset function, written as “reset.”

1 Overview

1.1 Introduction

The DSP56364 24-Bit Digital Signal Processor, a new audio digital signal processor based on the 24-bit DSP56300 architecture, is targeted to applications that require digital audio signal processing such as sound field processing, acoustic equalization and other digital audio algorithms.

The DSP56364 supports digital audio applications requiring sound field processing, acoustic equalization, and other digital audio algorithms. The DSP56364 uses the high performance, single-clock-per-cycle DSP56300 core family of programmable CMOS digital signal processors (DSPs) combined with the audio signal processing capability of the Freescale Symphony™ DSP family, as shown in Figure 1-1. This design provides a two-fold performance increase over Freescale's popular Symphony family of DSPs while retaining code compatibility. Significant architectural enhancements include a barrel shifter, 24-bit addressing, instruction cache, and direct memory access (DMA).

This document is intended to be used with the following Freescale publications:

- DSP56300 24-Bit Digital Signal Processor Family Manual, Freescale publication DSP56300FM.
- DSP56364 24-Bit Digital Signal Processor Technical Data Sheet, Freescale publication DSP56364.

The DSP56300 24-Bit Digital Signal Processor Family Manual, Freescale publication DSP56300FM provides a description of the components of the DSP56300 modular chassis which is common to all DSP56300 family processors and includes a detailed description of the instruction set. This document provides a detailed description of the core configuration, memory, and peripherals that are specific to the DSP56364. The electrical specifications, timings and packaging information can be found in the *DSP56364 Technical Data Sheet.*

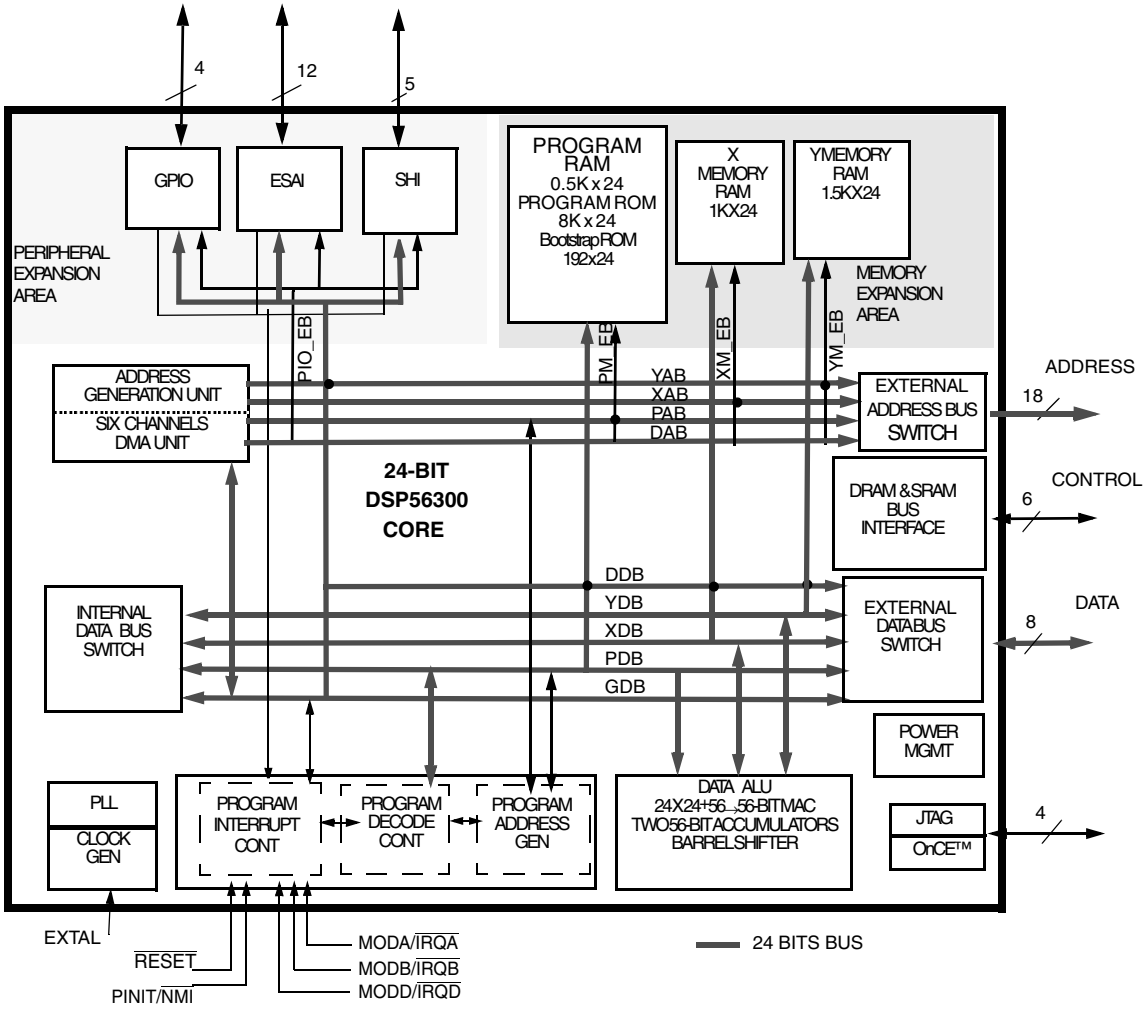


Figure 1-1 DSP56364 Block Diagram

1.2 Features

- DSP56300 modular chassis
  - 100 Million Instructions Per Second (MIPS) with an 100 MHz clock at 3.3V.
  - Object Code Compatible with the 56K core.
  - Data ALU with a 24 × 24 bit multiplier-accumulator and a 56-bit barrel shifter. 16-bit arithmetic support.
  - Program Control with position independent code support and instruction cache support.
  - Six-channel DMA controller.
  - PLL based clocking with a wide range of frequency multiplications (1 to 4096), predivider factors (1 to 16) and power saving clock divider (2^{i}: i = 0 to 7). Reduces clock noise.
  - Internal address tracing support and OnCE‰ for Hardware/Software debugging.
  - JTAG port.

  - Very low-power CMOS design, fully static design with operating frequencies down to DC.
  - STOP and WAIT low-power standby modes.
- On-chip Memory Configuration
  - 1.5K × 24 Bit Y-Data RAM.
  - 1K × 24 Bit X-Data RAM.
  - 8K × 24 Bit Program ROM.
  - 0.5K × 24 Bit Program RAM and 192 × 24 Bit Bootstrap ROM.
  - 0.75K × 24 Bit from Y Data RAM can be switched to Program RAM resulting in up to 1.25K × 24 Bit of Program RAM.
- Off-chip memory expansion
  - External Memory Expansion Port with 8-bit data bus.
  - Off-chip expansion up to 2 × 16M x 8-bit word of Data memory when using DRAM.
  - Off-chip expansion up to 2 × 256K x 8-bit word of Data memory when using SRAM.
  - Simultaneous glueless interface to SRAM and DRAM.
- Peripheral modules
  - Serial Audio Interface (ESAI): up to 4 receivers and up to 6 transmitters, master or slave. I^2S, Sony, AC97, network and other programmable protocols. Unused pins of ESAI may be used as GPIO lines.
  - Serial Host Interface (SHI): SPI and I^2C protocols, multi master capability, 10-word receive FIFO, support for 8, 16 and 24-bit words.
  - Four dedicated GPIO lines.
- 100-pin plastic TQFP package.

1.3 Audio Processor Architecture

This section defines the DSP56364 audio processor architecture. The audio processor is composed of the following units:

- The DSP56300 core is composed of the Data ALU, Address Generation Unit, Program Controller, Instruction-Cache Controller, DMA Controller, PLL-based clock oscillator, Memory Module Interface, Peripheral Module Interface and the On-Chip Emulator (OnCE). The DSP56300 core is described in the document *DSP56300 24-Bit Digital Signal Processor Family Manual, Motorola publication DSP56300FM*.
- Memory modules.
- Peripheral modules. The SHI, ESAI and GPIO peripheral are described in this document.

See Figure 1-1 for the block diagram of the DSP56364.

1.4 Core Description

The DSP56364 uses the DSP56300 core, a high-performance, single clock cycle per instruction engine that provides up to twice the performance of Freescale's popular DSP56000 core family while retaining code compatibility with it.

The DSP56300 core provides the following functional blocks:

- Data arithmetic logic unit (Data ALU)
- Address generation unit (AGU)
- Program control unit (PCU)
- Bus interface unit (BIU)
- DMA controller (with six channels)
- Instruction cache controller
- PLL-based clock oscillator
- OnCE module
- JTAG TAP
- Memory

The DSP56300 core family offers a new level of performance in speed and power, provided by its rich instruction set and low power dissipation, thus enabling a new generation of wireless, telecommunications, and multimedia products. Significant architectural enhancements to the DSP56300 core family include a barrel shifter, 24-bit addressing, an instruction cache, and direct memory access (DMA).

Core features are described fully in the *DSP56300 Family Manual*. Pinout, memory, and peripheral features are described in this manual.

1.5 DSP56300 Core Functional Blocks

1.5.1 Data ALU

The Data ALU performs all the arithmetic and logical operations on data operands in the DSP56300 core. The components of the Data ALU are as follows:

- Fully pipelined 24-bit × 24-bit parallel multiplier-accumulator (MAC)
- Bit field unit, comprising a 56-bit parallel barrel shifter (fast shift and normalization; bit stream generation and parsing)
- Conditional ALU instructions
- 24-bit or 16-bit arithmetic support under software control
- Four 24-bit input general purpose registers: X1, X0, Y1, and Y0
- Six Data ALU registers (A2, A1, A0, B2, B1, and B0) that are concatenated into two general purpose, 56-bit accumulators (A and B), accumulator shifters
- Two data bus shifter/limiter circuits

1.5.1.1 Data ALU Registers

The Data ALU registers can be read or written over the X memory data bus (XDB) and the Y memory data bus (YDB) as 24- or 48-bit operands (or as 16- or 32-bit operands in 16-bit arithmetic mode). The source operands for the Data ALU, which can be 24, 48, or 56 bits (16, 32, or 40 bits in 16-bit arithmetic mode), always originate from Data ALU registers. The results of all Data ALU operations are stored in an accumulator.

All the Data ALU operations are performed in two clock cycles in pipeline fashion so that a new instruction can be initiated in every clock, yielding an effective execution rate of one instruction per clock cycle. The destination of every arithmetic operation can be used as a source operand for the immediately following arithmetic operation without a time penalty (in other words, without a pipeline stall).

1.5.1.2 Multiplier-Accumulator (MAC)

The MAC unit comprises the main arithmetic processing unit of the DSP56300 core and performs all of the calculations on data operands. In the case of arithmetic instructions, the unit accepts as many as three input operands and outputs one 56-bit result of the following form- Extension:Most Significant Product:Least Significant Product (EXT:MSP:LSP).

The multiplier executes 24-bit × 24-bit, parallel, fractional multiplies, between two's-complement signed, unsigned, or mixed operands. The 48-bit product is right-justified and added to the 56-bit contents of either the A or B accumulator. A 56-bit result can be stored as a 24-bit operand. The LSP can either be truncated or rounded into the MSP. Rounding is performed if specified.

1.5.2 Address Generation Unit (AGU)

The AGU performs the effective address calculations using integer arithmetic necessary to address data operands in memory and contains the registers used to generate the addresses. It implements four types of arithmetic: linear, modulo, multiple wrap-around modulo, and reverse-carry. The AGU operates in parallel with other chip resources to minimize address-generation overhead.

The AGU is divided into two halves, each with its own Address ALU. Each Address ALU has four sets of register triplets, and each register triplet is composed of an address register, an offset register, and a modifier register. The two Address ALUs are identical. Each contains a 24-bit full adder (called an offset adder).

A second full adder (called a modulo adder) adds the summed result of the first full adder to a modulo value that is stored in its respective modifier register. A third full adder (called a reverse-carry adder) is also provided.

The offset adder and the reverse-carry adder are in parallel and share common inputs. The only difference between them is that the carry propagates in opposite directions. Test logic determines which of the three summed results of the full adders is output.

Each Address ALU can update one address register from its respective address register file during one instruction cycle. The contents of the associated modifier register specifies the type of arithmetic to be used in the address register update calculation. The modifier value is decoded in the Address ALU.

1.5.3 Program Control Unit (PCU)

The PCU performs instruction prefetch, instruction decoding, hardware DO loop control, and exception processing. The PCU implements a seven-stage pipeline and controls the different processing states of the DSP56300 core. The PCU consists of the following three hardware blocks:

- Program decode controller (PDC)
- Program address generator (PAG)
- Program interrupt controller (PIC)

The PDC decodes the 24-bit instruction loaded into the instruction latch and generates all signals necessary for pipeline control. The PAG contains all the hardware needed for program address generation, system stack, and loop control. The PIC arbitrates among all interrupt requests (internal interrupts, as well as the five external requests: $\overline{\text{IRQA}}$, $\overline{\text{IRQB}}$, $\overline{\text{IRQD}}$, and $\overline{\text{NMI}}$), and generates the appropriate interrupt vector address.

PCU features include the following:

- Position independent code support
- Addressing modes optimized for DSP applications (including immediate offsets)
- On-chip instruction cache controller
- On-chip memory-expandable hardware stack
- Nested hardware DO loops
- Fast auto-return interrupts

The PCU implements its functions using the following registers:

- PC—program counter register
- SR—Status register
- LA—loop address register
- LC—loop counter register
- VBA—vector base address register
- SZ—stack size register
- SP—stack pointer
- OMR—operating mode register
- SC—stack counter register

The PCU also includes a hardware system stack (SS).

1.5.4 Internal Buses

To provide data exchange between blocks, the following buses are implemented:

- Peripheral input/output expansion bus (PIO_EB) to peripherals
- Program memory expansion bus (PM_EB) to program memory
- X memory expansion bus (XM_EB) to X memory

- Y memory expansion bus (YM_EB) to Y memory
- Global data bus (GDB) between registers in the DMA, AGU, OnCE, PLL, BIU, and PCU as well as the memory-mapped registers in the peripherals
- DMA data bus (DDB) for carrying DMA data between memories and/or peripherals
- DMA address bus (DAB) for carrying DMA addresses to memories and peripherals
- Program Data Bus (PDB) for carrying program data throughout the core
- X memory Data Bus (XDB) for carrying X data throughout the core
- Y memory Data Bus (YDB) for carrying Y data throughout the core
- Program address bus (PAB) for carrying program memory addresses throughout the core
- X memory address bus (XAB) for carrying X memory addresses throughout the core
- Y memory address bus (YAB) for carrying Y memory addresses throughout the core

All internal buses on the DSP56300 family members are 24-bit buses. See Figure 1-1, DSP56364 block diagram.

1.5.5 Direct Memory Access (DMA)

The DMA block has the following features:

- Six DMA channels supporting internal and external accesses
- One-, two-, and three-dimensional transfers (including circular buffering)
- End-of-block-transfer interrupts
- Triggering from interrupt lines and all peripherals

1.5.6 PLL-based Clock Oscillator

The clock generator in the DSP56300 core is composed of two main blocks: the PLL, which performs clock input division, frequency multiplication, and skew elimination; and the clock generator (CLKGEN), which performs low-power division and clock pulse generation. PLL-based clocking:

- Allows change of low-power divide factor (DF) without loss of lock
- Provides output clock with skew elimination
- Provides a wide range of frequency multiplications (1 to 4096), predivider factors (1 to 16), and a power-saving clock divider (2^i: i = 0 to 7) to reduce clock noise

The PLL allows the processor to operate at a high internal clock frequency using a low frequency clock input. This feature offers two immediate benefits:

- A lower frequency clock input reduces the overall electromagnetic interference generated by a system.
- The ability to oscillate at different frequencies reduces costs by eliminating the need to add additional oscillators to a system.

1.5.7 JTAG TAP and OnCE Module

The DSP56300 core provides a dedicated user-accessible TAP fully compatible with the *IEEE 1149.1 Standard Test Access Port and Boundary Scan Architecture*. Problems associated with testing high-density circuit boards led to developing this standard under the sponsorship of the Test Technology Committee of IEEE and JTAG. The DSP56300 core implementation supports circuit-board test strategies based on this standard.

The test logic includes a TAP consisting of four dedicated signals, a 16-state controller, and three test data registers. A boundary scan register links all device signals into a single shift register. The test logic, implemented utilizing static logic design, is independent of the device system logic.

The OnCE module provides a nonintrusive means of interacting with the DSP56300 core and its peripherals so a user can examine registers, memory, or on-chip peripherals. This facilitates hardware and software development on the DSP56300 core processor. OnCE module functions are provided through the JTAG TAP signals.

1.6 Data and Program memory

The on-chip memory configuration of the DSP56364 is affected by the state of the MS (Memory Switch) control bit in the OMR register, and by the SC bit in the Status Register. Refer to Section 3, "Memory Configuration".

1.6.1 Reserved Memory Spaces

The reserved memory spaces should not be accessed by the user. They are reserved for future expansion.

1.6.2 Program ROM Area Reserved for Freescale Use

The last 128 words ($FF2F80-$FF2FFF) of the Program ROM are reserved for Freescale use. This memory area is reserved for use as expansion area for the bootstrap ROM as well as for testing purposes. Customer code should not use this area. The contents of this Program ROM segment is defined by the Bootstrap ROM source code in Appendix A, "Bootstrap ROM".

1.6.3 Bootstrap ROM

The 192-word Bootstrap ROM occupies locations $FF0000-$FF00BF. The bootstrap ROM is factory-programmed to perform the bootstrap operation following hardware reset. The contents of the Bootstrap ROM are defined by the Bootstrap ROM source code in Appendix A, "Bootstrap ROM".

1.6.4 Dynamic Memory Configuration Switching

The internal memory configuration is altered by re-mapping RAM modules from Y data memory into program memory space and vice-versa. The contents of the switched RAM modules are preserved.

The memory can be dynamically switched from one configuration to another by changing the MS bit in OMR. The address ranges that are directly affected by the switch operation are specified in Table 3-1 The

memory switch can be accomplished provided that the affected address ranges are not being accessed during the instruction cycle in which the switch operation takes place. Accordingly, the following condition must be observed for trouble-free dynamic switching:

NOTE

No accesses (including instruction fetches) to or from the affected address ranges in program and data memories are allowed during the switch cycle.

NOTE

The switch cycle actually occurs 3 instruction cycles after the instruction that modifies the MS bit.

Any sequence that complies with the switch condition is valid. For example, if the program flow executes in the address range that is not affected by the switch, the switch condition can be met very easily. In this case a switch can be accomplished by just changing the MS bit in OMR in the regular program flow, assuming no accesses to the affected address ranges of the data memory occur up to 3 instructions after the instruction that changes the OMR bit. Special care should be taken in relation to the interrupt vector routines since an interrupt could cause the DSP to fetch instructions out of sequence and might violate the switch condition.

Special attention should be given when running a memory switch routine using the OnCE™ port. Running the switch routine in Trace mode, for example, can cause the switch to complete after the MS bit change while the DSP is in Debug mode. As a result, subsequent instructions might be fetched according to the new memory configuration (after the switch), and thus might execute improperly.

1.6.5 External Memory Support

The DSP56364 does not support the SSRAM memory type. It does support SRAM and DRAM as indicated in the *DSP56300 24-Bit Digital Signal Processor Family Manual, Motorola publication DSP56300FM*. Note that the DSP56364 has only an 8-bit data bus. This means that no instruction fetches from external memory are possible, and care should be taken to ensure that no program memory instruction fetch access occurs in the external memory space. The DMA may be used to automatically pack and unpack 24-bit data into the 8-bit wide external memory, during DMA data transfers.

Also, care should be taken when accessing external memory to ensure that the necessary address lines are available. For example, when using glueless SRAM interfacing, it is possible to directly address 2^{19} memory locations (512K) when using the 18 address lines and the two programmable address attribute lines. Using DRAM access mode, the full 16M addressing range may be used.

1.7 Internal I/O Memory Map

The DSP56364 on-chip peripheral modules have their register files programmed to the addresses in the internal X-I/O memory range (the top 128 locations of the X data memory space). See Section 3, “Memory Configuration.”

1.8 Status Register (SR)

Refer to the DSP56300 24-Bit Digital Signal Processor Family Manual, Freescale publication DSP56300FM/AD for a description of the Status Register bits.

The Cache Enable bit (Bit 19) in the Status Register must be kept cleared since the DSP56364 does not have an on-chip instruction cache.

2 Signal/Connection Descriptions

2.1 Signal Groupings

The input and output signals of the DSP56364 are organized into functional groups, which are listed in Table 2-1 and illustrated in Figure 2-1.

The DSP56364 is operated from a 3.3 V supply; however, some of the inputs can tolerate 5 V. A special notice for this feature is added to the signal descriptions of those inputs.

Table 2-1 DSP56364 Functional Signal Groupings

Functional Group		Number of Signals	Detailed Description
Power (V_{CC})		18	Table 2-2
Ground (GND)		14	Table 2-3
Clock and PLL		3	Table 2-4
Address bus	Port A[1]	18	Table 2-5
Data bus		8	Table 2-6
Bus control		6	Table 2-7
Interrupt and mode control		4	Table 2-8
General Purpose I/O	Port B[2]	4	Table 2-12
SHI		5	Table 2-9
ESAI	Port C[3]	12	Table 2-10
JTAG/OnCE Port		4	Table 2-11

1 Port A is the external memory interface port, including the external address bus, data bus, and control signals.

2 Port B signals are the GPIO signals.

3 Port C signals are the ESAI port signals multiplexed with the GPIO signals.

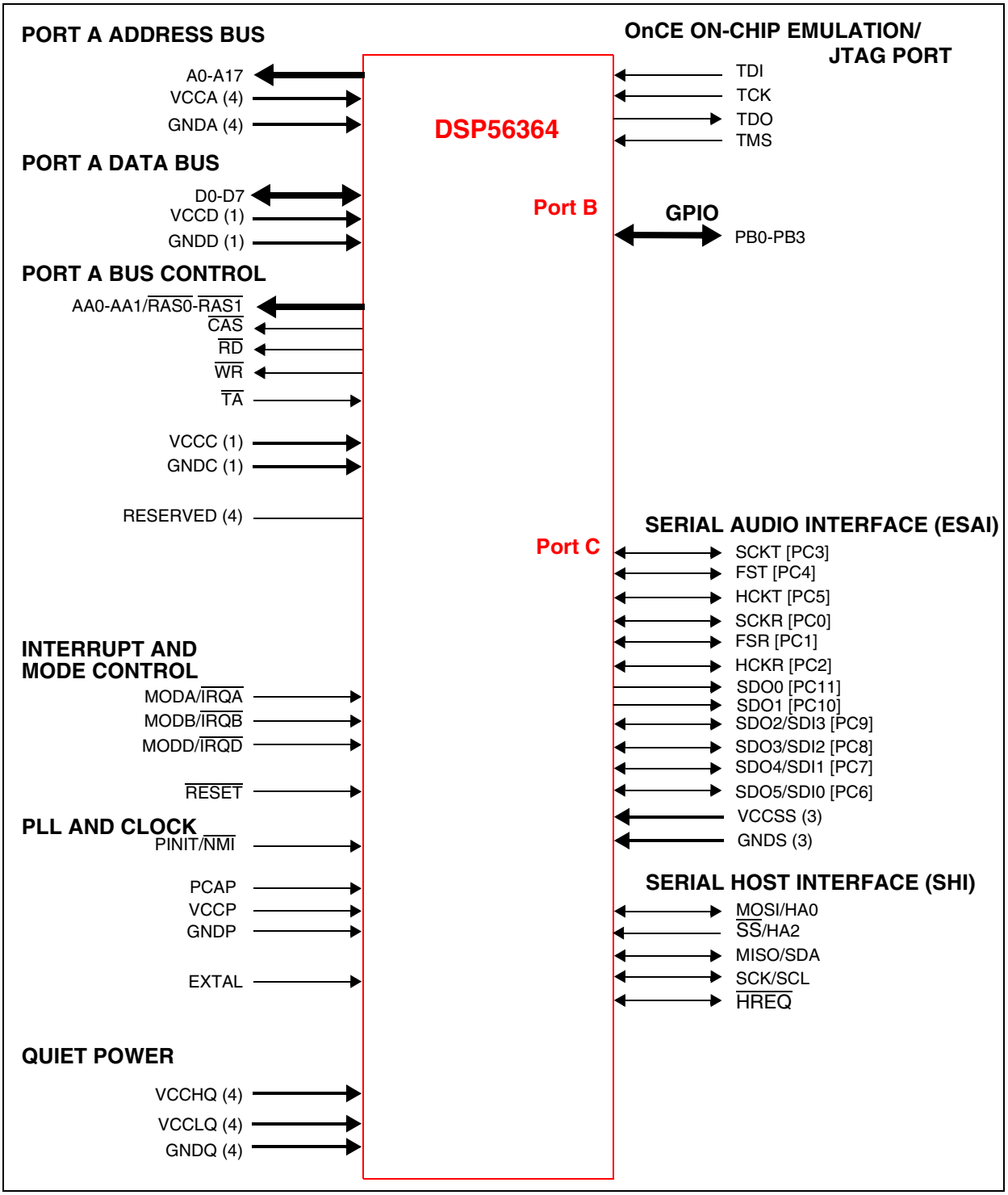


Figure 2-1 Signals Identified by Functional Group

2.2 Power

Table 2-2 Power Inputs

Power Name	Description
V_{CCP}	**PLL Power**—V_{CCP} is V_{CC} dedicated for PLL use. The voltage should be well-regulated and the input should be provided with an extremely low impedance path to the V_{CC} power rail. There is one V_{CCP} input.
V_{CCQL} (4)	**Quiet Core (Low) Power—**V_{CCQL} is an isolated power for the internal processing logic. This input must be tied externally to all other chip power inputs. The user must provide adequate external decoupling capacitors. There are four V_{CCQL} inputs.
V_{CCQH} (4)	**Quiet External (High) Power**—V_{CCQH} is a quiet power source for I/O lines. This input must be tied externally to all other chip power inputs. The user must provide adequate decoupling capacitors. There are four V_{CCQH} inputs.
V_{CCA} (4)	**Address Bus Power**—V_{CCA} is an isolated power for sections of the address bus I/O drivers. This input must be tied externally to all other chip power inputs. The user must provide adequate external decoupling capacitors. There are four V_{CCA} inputs.
V_{CCD} (1)	**Data Bus Power**—V_{CCD} is an isolated power for sections of the data bus I/O drivers. This input must be tied externally to all other chip power inputs. The user must provide adequate external decoupling capacitors. There is one V_{CCD} inputs.
V_{CCC} (1)	**Bus Control Power**—V_{CCC} is an isolated power for the bus control I/O drivers. This input must be tied externally to all other chip power inputs. The user must provide adequate external decoupling capacitors. There is one V_{CCC} inputs.
V_{CCS} (3)	**SHI and ESAI** —V_{CCS} is an isolated power for the SHI and ESAI. This input must be tied externally to all other chip power inputs$_L$. The user must provide adequate external decoupling capacitors. There are three V_{CCS} inputs.

2.3 Ground

Table 2-3 Grounds

Ground Name	Description
GND_P	**PLL Ground**—GND_P is ground-dedicated for PLL use. The connection should be provided with an extremely low-impedance path to ground. V_{CCP} should be bypassed to GND_P by a 0.47 µF capacitor located as close as possible to the chip package. There is one GND_P connection.
GND_Q (4)	**Quiet Ground**—GND_Q is an isolated ground for the internal processing logic. This connection must be tied externally to all other chip ground connections. The user must provide adequate external decoupling capacitors. There are four GND_Q connections.
GND_A (4)	**Address Bus Ground**—GND_A is an isolated ground for sections of the address bus I/O drivers. This connection must be tied externally to all other chip ground connections. The user must provide adequate external decoupling capacitors. There are four GND_A connections.
GND_D (1)	**Data Bus Ground**—GND_D is an isolated ground for sections of the data bus I/O drivers. This connection must be tied externally to all other chip ground connections. The user must provide adequate external decoupling capacitors. There is one GND_D connections.

Table 2-3 Grounds (continued)

Ground Name	Description
GND_C (1)	**Bus Control Ground**—GND_C is an isolated ground for the bus control I/O drivers. This connection must be tied externally to all other chip ground connections. The user must provide adequate external decoupling capacitors. There is one GND_C connections.
GND_S (3)	**SHI and ESAI** —GND_S is an isolated ground for the SHI and ESAI. This connection must be tied externally to all other chip ground connections. The user must provide adequate external decoupling capacitors. There are three GND_S connections.

2.4 Clock and PLL

Table 2-4 Clock and PLL Signals

Signal Name	Signal Type	State during Reset	Signal Description
EXTAL	Input	Input	**External Clock Input**—An external clock source must be connected to EXTAL in order to supply the clock to the internal clock generator and PLL.
PCAP	Input	Input	**PLL Capacitor**—PCAP is an input connecting an off-chip capacitor to the PLL filter. Connect one capacitor terminal to PCAP and the other terminal to V_{CCP}. If the PLL is not used, PCAP may be tied to V_{CC}, GND, or left floating.
PINIT/$\overline{\text{NMI}}$	Input	Input	**PLL Initial/Nonmaskable Interrupt**—During assertion of $\overline{\text{RESET}}$, the value of PINIT/$\overline{\text{NMI}}$ is written into the PLL Enable (PEN) bit of the PLL control register, determining whether the PLL is enabled or disabled. After $\overline{\text{RESET}}$ de assertion and during normal instruction processing, the PINIT/$\overline{\text{NMI}}$ Schmitt-trigger input is a negative-edge-triggered nonmaskable interrupt (NMI) request internally synchronized to internal system clock. *This input is 5 V tolerant.*

2.5 External Memory Expansion Port (Port A)

When the DSP56364 enters a low-power standby mode (stop or wait), it tri-states the relevant port A signals: D0–D7, AA0, AA1, $\overline{\text{RD}}$, $\overline{\text{WR}}$, $\overline{\text{CAS}}$.

2.5.1 External Address Bus

Table 2-5 External Address Bus Signals

Signal Name	Signal Type	State during Reset	Signal Description
A0–A17	Output	Keeper active	**Address Bus**—A0–A17 are active-high outputs that specify the address for external program and data memory accesses. Otherwise, the signals are kept to their previous values by internal weak keepers. To minimize power dissipation, A0–A17 do not change state when external memory spaces are not being accessed.

2.5.2 External Data Bus

Table 2-6 External Data Bus Signals

Signal Name	Signal Type	State during Reset	Signal Description
D0–D7	Input/Output	Tri-stated	**Data Bus**—D0–D7 are active-high, bidirectional input/outputs that provide the bidirectional data bus for external program and data memory accesses. D0–D7 are tri-stated during hardware reset and when the DSP is in the stop or wait low-power standby mode.

2.5.3 External Bus Control

Table 2-7 External Bus Control Signals

Signal Name	Signal Type	State during Reset	Signal Description
AA0–AA1/$\overline{RAS0}$–$\overline{RAS1}$	Output	Tri-stated	**Address Attribute or Row Address Strobe**—When defined as AA, these signals can be used as chip selects or additional address lines. When defined as $\overline{RAS}$, these signals can be used as $\overline{RAS}$ for DRAM interface. These signals are tri-statable outputs with programmable polarity. These signals are tri-stated during hardware reset and when the DSP is in the stop or wait low-power standby mode.
CAS	Output	Tri-stated	**Column Address Strobe**— $\overline{CAS}$ is an active-low output used by DRAM to strobe the column address. This signal is tri-stated during hardware reset and when the DSP is in the stop or wait low-power standby mode.

Table 2-7 External Bus Control Signals (continued)

Signal Name	Signal Type	State during Reset	Signal Description
RD	Output	Tri-stated	**Read Enable**—$\overline{RD}$ is an active-low output that is asserted to read external memory on the data bus. This signal is tri-stated during hardware reset and when the DSP is in the stop or wait low-power standby mode.
WR	Output	Tri-stated	**Write Enable**— $\overline{WR}$ is an active-low output that is asserted to write external memory on the data bus. This signal is tri-stated during hardware reset and when the DSP is in the stop or wait low-power standby mode.
TA	Input	Ignored Input	**Transfer Acknowledge**—If there is no external bus activity, the $\overline{TA}$ input is ignored. The $\overline{TA}$ input is a data transfer acknowledge (DTACK) function that can extend an external bus cycle indefinitely. Any number of wait states (1, 2. . .infinity) may be added to the wait states inserted by the BCR by keeping $\overline{TA}$ deasserted. In typical operation, $\overline{TA}$ is deasserted at the start of a bus cycle, is asserted to enable completion of the bus cycle, and is deasserted before the next bus cycle. The current bus cycle completes one clock period after $\overline{TA}$ is asserted synchronous to the internal system clock. The number of wait states is determined by the $\overline{TA}$ input or by the bus control register (BCR), whichever is longer. The BCR can be used to set the minimum number of wait states in external bus cycles. In order to use the $\overline{TA}$ functionality, the BCR must be programmed to at least one wait state. A zero wait state access cannot be extended by $\overline{TA}$ deassertion, otherwise improper operation may result. $\overline{TA}$ can operate synchronously or asynchronously, depending on the setting of the TAS bit in the operating mode register (OMR). $\overline{TA}$ functionality may not be used while performing DRAM type accesses, otherwise improper operation may result.

2.6 Interrupt and Mode Control

The interrupt and mode control signals select the chip's operating mode as it comes out of hardware reset. After $\overline{RESET}$ is deasserted, these inputs are hardware interrupt request lines.

Table 2-8 Interrupt and Mode Control

Signal Name	Signal Type	State during Reset	Signal Description
MODA/$\overline{\text{IRQA}}$	Input	Input	**Mode Select A/External Interrupt Request A—**MODA/$\overline{\text{IRQA}}$ is an active-low Schmitt-trigger input, internally synchronized to the internal system clock. MODA/$\overline{\text{IRQA}}$ selects the initial chip operating mode during hardware reset and becomes a level-sensitive or negative-edge-triggered, maskable interrupt request input during normal instruction processing. MODA, MODB, and MODD select one of 8 initial chip operating modes, latched into the OMR when the $\overline{\text{RESET}}$ signal is deasserted. If $\overline{\text{IRQA}}$ is asserted synchronous to the internal system clock, multiple processors can be re synchronized using the WAIT instruction and asserting $\overline{\text{IRQA}}$ to exit the wait state. If the processor is in the stop standby state and $\overline{\text{IRQA}}$ is asserted, the processor will exit the stop state. *This input is 5 V tolerant.*
MODB/$\overline{\text{IRQB}}$	Input	Input	**Mode Select B/External Interrupt Request B—**MODB/$\overline{\text{IRQB}}$ is an active-low Schmitt-trigger input, internally synchronized to the internal system clock. MODB/$\overline{\text{IRQB}}$ selects the initial chip operating mode during hardware reset and becomes a level-sensitive or negative-edge-triggered, maskable interrupt request input during normal instruction processing. MODA, MODB, and MODD select one of 8 initial chip operating modes, latched into OMR when the $\overline{\text{RESET}}$ signal is deasserted. If $\overline{\text{IRQB}}$ is asserted synchronous to the internal system clock, multiple processors can be re-synchronized using the WAIT instruction and asserting $\overline{\text{IRQB}}$ to exit the wait state. *This input is 5 V tolerant.*
MODD/$\overline{\text{IRQD}}$	Input	Input	**Mode Select D/External Interrupt Request D**—MODD/$\overline{\text{IRQD}}$ is an active-low Schmitt-trigger input, internally synchronized to the internal system clock. MODD/$\overline{\text{IRQD}}$ selects the initial chip operating mode during hardware reset and becomes a level-sensitive or negative-edge-triggered, maskable interrupt request input during normal instruction processing. MODA, MODB, and MODD select one of 8 initial chip operating modes, latched into OMR when the $\overline{\text{RESET}}$ signal is deasserted. If $\overline{\text{IRQD}}$ is asserted synchronous to the internal system clock, multiple processors can be re synchronized using the WAIT instruction and asserting $\overline{\text{IRQD}}$ to exit the wait state. *This input is 5 V tolerant.*
RESET	Input	Input	**Reset—**$\overline{\text{RESET}}$ is an active-low, Schmitt-trigger input. When asserted, the chip is placed in the reset state and the internal phase generator is reset. The Schmitt-trigger input allows a slowly rising input (such as a capacitor charging) to reset the chip reliably. When the $\overline{\text{RESET}}$ signal is deasserted, the initial chip operating mode is latched from the MODA, MODB, and MODD inputs. The $\overline{\text{RESET}}$ signal must be asserted during power up. A stable EXTAL signal must be supplied before deassertion of $\overline{\text{RESET}}$. *This input is 5 V tolerant.*

2.7 Serial Host Interface

The SHI has five I/O signals that can be configured to allow the SHI to operate in either SPI or I^2C mode

Table 2-9 Serial Host Interface Signals

Signal Name	Signal Type	State during Reset	Signal Description
SCK	Input or output	Tri-stated	**SPI Serial Clock**—The SCK signal is an output when the SPI is configured as a master and a Schmitt-trigger input when the SPI is configured as a slave. When the SPI is configured as a master, the SCK signal is derived from the internal SHI clock generator. When the SPI is configured as a slave, the SCK signal is an input, and the clock signal from the external master synchronizes the data transfer. The SCK signal is ignored by the SPI if it is defined as a slave and the slave select ($\overline{SS}$) signal is not asserted. In both the master and slave SPI devices, data is shifted on one edge of the SCK signal and is sampled on the opposite edge where data is stable. Edge polarity is determined by the SPI transfer protocol.
SCL	Input or output		**I^2C Serial Clock**—SCL carries the clock for I^2C bus transactions in the I^2C mode. SCL is a Schmitt-trigger input when configured as a slave and an open-drain output when configured as a master. SCL should be connected to V_{CC} through a pull-up resistor.
			This signal is tri-stated during hardware, software, and individual reset. Thus, there is no need for an external pull-up in this state. This input is 5 V tolerant.
MISO	Input or output	Tri-stated	**SPI Master-In-Slave-Out**—When the SPI is configured as a master, MISO is the master data input line. The MISO signal is used in conjunction with the MOSI signal for transmitting and receiving serial data. This signal is a Schmitt-trigger input when configured for the SPI Master mode, an output when configured for the SPI Slave mode, and tri-stated if configured for the SPI Slave mode when $\overline{SS}$ is deasserted. An external pull-up resistor is not required for SPI operation.
SDA	Input or open-drain output		**I^2C Data and Acknowledge**—In I^2C mode, SDA is a Schmitt-trigger input when receiving and an open-drain output when transmitting. SDA should be connected to V_{CC} through a pull-up resistor. SDA carries the data for I^2C transactions. The data in SDA must be stable during the high period of SCL. The data in SDA is only allowed to change when SCL is low. When the bus is free, SDA is high. The SDA line is only allowed to change during the time SCL is high in the case of start and stop events. A high-to-low transition of the SDA line while SCL is high is a unique situation, and is defined as the start event. A low-to-high transition of SDA while SCL is high is a unique situation defined as the stop event.
			This signal is tri-stated during hardware, software, and individual reset. Thus, there is no need for an external pull-up in this state. This input is 5 V tolerant.

Table 2-9 Serial Host Interface Signals (continued)

Signal Name	Signal Type	State during Reset	Signal Description
MOSI	Input or output	Tri-stated	**SPI Master-Out-Slave-In**—When the SPI is configured as a master, MOSI is the master data output line. The MOSI signal is used in conjunction with the MISO signal for transmitting and receiving serial data. MOSI is the slave data input line when the SPI is configured as a slave. This signal is a Schmitt-trigger input when configured for the SPI Slave mode.
HA0	Input		**I^2C Slave Address 0**—This signal uses a Schmitt-trigger input when configured for the I^2C mode. When configured for I^2C slave mode, the HA0 signal is used to form the slave device address. HA0 is ignored when configured for the I^2C master mode.
			This signal is tri-stated during hardware, software, and individual reset. Thus, there is no need for an external pull-up in this state. *This input is 5 V tolerant.*
SS	Input	Input	**SPI Slave Select**—This signal is an active low Schmitt-trigger input when configured for the SPI mode. When configured for the SPI Slave mode, this signal is used to enable the SPI slave for transfer. When configured for the SPI master mode, this signal should be kept deasserted (pulled high). If it is asserted while configured as SPI master, a bus error condition is flagged. If $\overline{SS}$ is deasserted, the SHI ignores SCK clocks and keeps the MISO output signal in the high-impedance state.
HA2			**I^2C Slave Address 2**—This signal uses a Schmitt-trigger input when configured for the I^2C mode. When configured for the I^2C Slave mode, the HA2 signal is used to form the slave device address. HA2 is ignored in the I^2C master mode.
			This signal is tri-stated during hardware, software, and individual reset. Thus, there is no need for an external pull-up in this state. *This input is 5 V tolerant.*
HREQ	Input or Output	Tri-stated	**Host Request**—This signal is an active low Schmitt-trigger input when configured for the master mode but an active low output when configured for the slave mode.
			When configured for the slave mode, $\overline{HREQ}$ is asserted to indicate that the SHI is ready for the next data word transfer and deasserted at the first clock pulse of the new data word transfer. When configured for the master mode, $\overline{HREQ}$ is an input. When asserted by the external slave device, it will trigger the start of the data word transfer by the master. After finishing the data word transfer, the master will await the next assertion of $\overline{HREQ}$ to proceed to the next transfer.
			This signal is tri-stated during hardware, software, personal reset, or when the HREQ1–HREQ0 bits in the HCSR are cleared. There is no need for external pull-up in this state. *This input is 5 V tole*rant.

2.8 Enhanced Serial Audio Interface

Table 2-10 Enhanced Serial Audio Interface Signals

Signal Name	Signal Type	State during Reset	Signal Description
HCKR	Input or output	GPIO disconnected	**High Frequency Clock for Receiver**—When programmed as an input, this signal provides a high frequency clock source for the ESAI receiver as an alternate to the DSP core clock. When programmed as an output, this signal can serve as a high-frequency sample clock (e.g., for external digital to analog converters [DACs]) or as an additional system clock.
PC2	Input, output, or disconnected		**Port C 2**—When the ESAI is configured as GPIO, this signal is individually programmable as input, output, or internally disconnected. The default state after reset is GPIO disconnected. *This input is 5 V tolerant.*
HCKT	Input or output	GPIO disconnected	**High Frequency Clock for Transmitter**—When programmed as an input, this signal provides a high frequency clock source for the ESAI transmitter as an alternate to the DSP core clock. When programmed as an output, this signal can serve as a high frequency sample clock (e.g., for external DACs) or as an additional system clock.
PC5	Input, output, or disconnected		**Port C 5**—When the ESAI is configured as GPIO, this signal is individually programmable as input, output, or internally disconnected. The default state after reset is GPIO disconnected. This input is 5 V tolerant.
FSR	Input or output	GPIO disconnected	**Frame Sync for Receiver**—This is the receiver frame sync input/output signal. In the asynchronous mode (SYN=0), the FSR pin operates as the frame sync input or output used by all the enabled receivers. In the synchronous mode (SYN=1), it operates as either the serial flag 1 pin (TEBE=0), or as the transmitter external buffer enable control (TEBE=1, RFSD=1). When this pin is configured as serial flag pin, its direction is determined by the RFSD bit in the RCCR register. When configured as the output flag OF1, this pin will reflect the value of the OF1 bit in the SAICR register, and the data in the OF1 bit will show up at the pin synchronized to the frame sync in normal mode or the slot in network mode. When configured as the input flag IF1, the data value at the pin will be stored in the IF1 bit in the SAISR register, synchronized by the frame sync in normal mode or the slot in network mode.
PC1	Input, output, or disconnected		**Port C 1**—When the ESAI is configured as GPIO, this signal is individually programmable as input, output, or internally disconnected. The default state after reset is GPIO disconnected. This input is 5 V tolerant.

Table 2-10 Enhanced Serial Audio Interface Signals (continued)

Signal Name	Signal Type	State during Reset	Signal Description
FST	Input or output	GPIO disconnected	**Frame Sync for Transmitter**—This is the transmitter frame sync input/output signal. For synchronous mode, this signal is the frame sync for both transmitters and receivers. For asynchronous mode, FST is the frame sync for the transmitters only. The direction is determined by the transmitter frame sync direction (TFSD) bit in the ESAI transmit clock control register (TCCR).
PC4	Input, output, or disconnected		**Port C 4**—When the ESAI is configured as GPIO, this signal is individually programmable as input, output, or internally disconnected. The default state after reset is GPIO disconnected. This input is 5 V tolerant.
SCKR	Input or output	GPIO disconnected	**Receiver Serial Clock**—SCKR provides the receiver serial bit clock for the ESAI. The SCKR operates as a clock input or output used by all the enabled receivers in the asynchronous mode (SYN=0), or as serial flag 0 pin in the synchronous mode (SYN=1). When this pin is configured as serial flag pin, its direction is determined by the RCKD bit in the RCCR register. When configured as the output flag OF0, this pin will reflect the value of the OF0 bit in the SAICR register, and the data in the OF0 bit will show up at the pin synchronized to the frame sync in normal mode or the slot in network mode. When configured as the input flag IF0, the data value at the pin will be stored in the IF0 bit in the SAISR register, synchronized by the frame sync in normal mode or the slot in network mode.
PC0	Input, output, or disconnected		**Port C 0**—When the ESAI is configured as GPIO, this signal is individually programmable as input, output, or internally disconnected. The default state after reset is GPIO disconnected. This input is 5 V tolerant.
SCKT	Input or output	GPIO disconnected	**Transmitter Serial Clock**—This signal provides the serial bit rate clock for the ESAI. SCKT is a clock input or output used by all enabled transmitters and receivers in synchronous mode, or by all enabled transmitters in asynchronous mode.
PC3	Input, output, or disconnected		**Port C 3**—When the ESAI is configured as GPIO, this signal is individually programmable as input, output, or internally disconnected. The default state after reset is GPIO disconnected. This input is 5 V tolerant.

Table 2-10 Enhanced Serial Audio Interface Signals (continued)

Signal Name	Signal Type	State during Reset	Signal Description
SDO5	Output	GPIO disconnected	**Serial Data Output 5**—When programmed as a transmitter, SDO5 is used to transmit data from the TX5 serial transmit shift register.
SDI0	Input		**Serial Data Input 0**—When programmed as a receiver, SDI0 is used to receive serial data into the RX0 serial receive shift register.
PC6	Input, output, or disconnected		**Port C 6**—When the ESAI is configured as GPIO, this signal is individually programmable as input, output, or internally disconnected. The default state after reset is GPIO disconnected. This input is 5 V tolerant.
SDO4	Output	GPIO disconnected	**Serial Data Output 4**—When programmed as a transmitter, SDO4 is used to transmit data from the TX4 serial transmit shift register.
SDI1	Input		**Serial Data Input 1**—When programmed as a receiver, SDI1 is used to receive serial data into the RX1 serial receive shift register.
PC7	Input, output, or disconnected		**Port C 7**—When the ESAI is configured as GPIO, this signal is individually programmable as input, output, or internally disconnected. The default state after reset is GPIO disconnected. This input is 5 V tolerant.
SDO3	Output	GPIO disconnected	**Serial Data Output 3**—When programmed as a transmitter, SDO3 is used to transmit data from the TX3 serial transmit shift register.
SDI2	Input		**Serial Data Input 2**—When programmed as a receiver, SDI2 is used to receive serial data into the RX2 serial receive shift register.
PC8	Input, output, or disconnected		**Port C 8**—When the ESAI is configured as GPIO, this signal is individually programmable as input, output, or internally disconnected. The default state after reset is GPIO disconnected. This input is 5 V tolerant.
SDO2	Output	GPIO disconnected	**Serial Data Output 2**—When programmed as a transmitter, SDO2 is used to transmit data from the TX2 serial transmit shift register
SDI3	Input		**Serial Data Input 3**—When programmed as a receiver, SDI3 is used to receive serial data into the RX3 serial receive shift register.
PC9	Input, output, or disconnected		**Port C 9**—When the ESAI is configured as GPIO, this signal is individually programmable as input, output, or internally disconnected. The default state after reset is GPIO disconnected. This input is 5 V tolerant.

Table 2-10 Enhanced Serial Audio Interface Signals (continued)

Signal Name	Signal Type	State during Reset	Signal Description
SDO1 PC10	Output Input, output, or disconnected	GPIO disconnected	**Serial Data Output 1**—SDO1 is used to transmit data from the TX1 serial transmit shift register. **Port C 10**—When the ESAI is configured as GPIO, this signal is individually programmable as input, output, or internally disconnected. The default state after reset is GPIO disconnected. This input is 5 V tolerant.
SDO0 PC11	Output Input, output, or disconnected	GPIO disconnected	**Serial Data Output 0**—SDO0 is used to transmit data from the TX0 serial transmit shift register. **Port C 11**—When the ESAI is configured as GPIO, this signal is individually programmable as input, output, or internally disconnected. The default state after reset is GPIO disconnected. This input is 5 V tolerant.

2.9 JTAG/OnCE Interface

Table 2-11 JTAG/OnCE Interface

Signal Name	Signal Type	State during Reset	Signal Description
TCK	Input	Input	**Test Clock**—TCK is a test clock input signal used to synchronize the JTAG test logic. It has an internal pull-up resistor. *This input is 5 V tolerant.*
TDI	Input	Input	**Test Data Input**—TDI is a test data serial input signal used for test instructions and data. TDI is sampled on the rising edge of TCK and has an internal pull-up resistor. This input is 5 V tolerant.
TDO	Output	Tri-stated	**Test Data Output**—TDO is a test data serial output signal used for test instructions and data. TDO is tri-statable and is actively driven in the shift-IR and shift-DR controller states. TDO changes on the falling edge of TCK.
TMS	Input	Input	**Test Mode Select**—TMS is an input signal used to sequence the test controller's state machine. TMS is sampled on the rising edge of TCK and has an internal pull-up resistor. This input is 5 V tolerant.

2.10 GPIO Signals

Table 2-12 GPIO Signals

Signal Name	Signal Type	State during Reset	Signal Description
GPIO0-GPIO3	Input, output or disconnected	disconnected	GPIO0-3- The General Purpose I/O pins are used for control and handshake functions between the DSP and external circuitry. Each Port B GPIO pin may be individually programmed as an input, output or disconnected

3 Memory Configuration

3.1 Memory Spaces

The DSP56364 provides the following three independent memory spaces:

- Program
- X data
- Y data

Each memory space uses (by default) 18 external address lines for addressing, allowing access to 256K of external memory when using the SRAM operating mode, and 16 M when using the DRAM operating mode. Program and data word length is 24 bits, and internal memory uses 24-bit addressing.

The DSP56364 provides a 16-bit compatibility mode that effectively uses 16-bit addressing for each memory space, allowing access to 64K of memory for each. This mode puts zeros in the most significant byte of the usual (24-bit) program and data word and ignores the zeroed byte, thus effectively using 16-bit program and data words. The 16-bit Compatibility mode allows the DSP56364 to use 56000 object code without change (thus minimizing system cost for applications that use the smaller address space). See the *DSP56300 Family Manual*, **Section 6.4**, for further information.

3.1.1 Program Memory Space

Program memory space consists of the following:

- Internal program memory, consisting of program RAM, 0.5K by default, and program ROM, 8K x 24-bit
- Bootstrap program ROM (192 x 24-bit)

3.1.1.1 Program RAM

On-chip program RAM occupies the lowest 0.5K (default) or 1.25K locations in the program memory space (depending on the setting of the MS bit). The program RAM default organization is 2 banks of 256 24-bit words (0.5K). The upper 3 banks of Y data RAM can be configured as program RAM by setting the MS bit.

3.1.1.2 Program ROM

The program ROM contains customer-supplied code. For further information on supplying code for a customized DSP56364 program ROM, please contact your Freescale regional sales office.

The last 128 words ($FF2F80-$FF2FFF) of the program ROM are reserved for Freescale use. This memory area is reserved for use as expansion area for the bootstrap ROM as well as for testing purposes.

Customer code should not use this area. The contents of this program ROM segment is defined by the bootstrap ROM source code in Appendix A, "Bootstrap ROM".

3.1.1.3 Bootstrap ROM

The bootstrap code is accessed at addresses $FF0000 to $FFF0BF (192 words) in program memory space. The bootstrap ROM is factory-programmed to perform the bootstrap operation following hardware reset. The bootstrap ROM can not be accessed in 16-bit address compatibility mode. See Appendix A for a complete listing of the bootstrap code.

3.1.1.4 Reserved Program Memory Locations

Program memory space at locations $FF00C0 to $FF0FFF and $FF3000 to $FFFFFF is reserved and should not be accessed.

3.1.2 Data Memory Spaces

Data memory space is divided into X data memory and Y data memory to match the natural partitioning of DSP algorithms. The data memory partitioning allows the DSP56364 to feed two operands to the Data ALU simultaneously, enabling it to perform a multiply-accumulate operation in one clock cycle.

X and Y data memory space are similar in structure and functionality, but there are two differences between them. First, part of Y data RAM may be switched over to program RAM, while X data RAM is fixed in size. Second, the upper 128 words of each space are reserved for different uses. The upper 128 words of X data memory are reserved for internal I/O. It is suggested that the programmer reserve the upper 128 words of Y data memory for external I/O. (For further information, see Section 3.1.2.1, "X Data Memory Space" and Section 3.1.2.3, "Y Data Memory Space")

X and Y data memory space each consist of the following:

- Internal data RAM memory (X data RAM (1K), and Y data RAM (default size is 1.5K, but 0.75K of Y data RAM can be switched to program RAM)
- (Optionally) Off-chip memory expansion (up to 256K in the 24-bit address mode and 64K in the 16-bit address mode).

3.1.2.1 X Data Memory Space

The on-chip peripheral registers and some of the DSP56364 core registers occupy the top 128 locations of X data memory ($FFFF80–$FFFFFF in the 24-bit address mode or $FF80–$FFFF in the 16-bit address mode). This area is called X-I/O space, and it can be accessed by MOVE and MOVEP instructions and by bit oriented instructions (BCHG, BCLR, BSET, BTST, BRCLR, BRSET, BSCLR, BSSET, JCLR, JSET, JSCLR, and JSSET). For a listing of the contents of this area, see the programming sheets in Section C.2, "Programming Sheets".

The reserved X memory space from $FF0000 to $FFEFFF should not be accessed.

3.1.2.2 X Data RAM

The on-chip X data RAM consists of 24-bit wide, high-speed, internal static RAM occupying 1K locations in the X memory space. The X data RAM organization is 4 banks of 256 24-bit words.

3.1.2.3 Y Data Memory Space

The off-chip peripheral registers should be mapped into the top 128 locations of Y data memory ($FFFF80–$FFFFFF in the 24-bit address mode or $FF80–$FFFF in the 16-bit address mode) to take advantage of the move peripheral data (MOVEP) instruction and the bit oriented instructions (BCHG, BCLR, BSET, BTST, BRCLR, BRSET, BSCLR, BSSET, JCLR, JSET, JSCLR, and JSSET).

The reserved Y memory space from $FF0000 to $FFEFFF should not be accessed.

3.1.2.4 Y Data RAM

The on-chip Y data RAM consists of 24-bit wide, high-speed, internal static RAM occupying 1.5K (default) or 0.75K locations in the Y memory space. The size of the Y data RAM is dependent on the setting of the MS bit (default: MS is cleared). The Y data RAM default organization is 6 banks of 256 24-bit words. Three banks of RAM may be switched to program RAM by setting the MS bit.

3.2 Memory Space Configuration

Memory space addressing is for 24-bit words by default. The DSP56364 switches to 16-bit address compatibility mode by setting the 16-bit compatibility (SC) bit in the SR.

Table 3-1 Memory Space Configuration Bit Settings for the DSP56364

Bit Abbreviation	Bit Name	Bit Location	Cleared = 0 Effect (Default)	Set = 1 Effect
SC	16-bit Compatibility	SR 13	16 M word address space (24-bit word)	64 K word address space (16-bit word)

Accessible external memory in the 24-bit mode is limited to a maximum of 512K when using the SRAM operating mode and to 16M when using the DRAM operating mode.

Memory maps for the different configurations are shown in Figure 3-1 to Figure 3-4.

3.3 Internal Memory Configuration

The following subsections discuss the internal memory configuration of the DSP56364. The size and location configurations for RAM and ROM for the DSP56364 are given below.

Table 3-2 Internal Memory Configurations

Bit Settings		Memory Sizes (24-bit words)				
MS	SC	Prog. RAM	Prog. ROM	Boot ROM	X Data RAM	Y Data RAM
0	0	0.5K	8K	192	1K	1.5K
1	0	1.25K	8K	192	1K	0.75K
0	1	0.5K	n.a.	n.a.	1K	1.5K
1	1	1.25K	n.a.	n.a.	1K	0.75K

Memory maps for the different configurations are shown in Figure 3-1 to Figure 3-4.

3.3.1 RAM Locations

The actual memory locations for program RAM and Y data RAM in their own memory space are determined by the MS bit. The memory location of X data RAM is independent of the MS bit. The addresses of the different RAMs are listed in Table 3-3.

Table 3-3 On-chip RAM Memory Locations

Bit Settings		RAM Memory Locations		
MS	SC	Program RAM	X Data RAM	Y Data RAM
0	X	$000000-$0001FF	$000000-$0003FF	$000000-$0005FF
1	X	$000000-$0004FF	$000000-$0003FF	$000000-$0002FF

3.3.2 ROM Locations

The actual memory locations for ROMs in their own memory space are fixed, but when the SC bit is set (i. e. the chip is in 16-bit mode), the program ROM and the bootstrap ROM are not accessible. ROM addresses are listed in Table 3-4.

Table 3-4 On-chip ROM Memory Locations

Bit Settings		ROM Memory Locations	
MS	SC	Program ROM	Boot ROM
X	0	$FF1000-$FF2FFF	$FF0000-$FF00BF
X	1	no access	no access

3.3.3 Dynamic Memory Configuration Switching

The internal memory configuration is altered by remapping RAM modules from Y data memory into program memory space and vice-versa. The contents of the switched RAM modules are preserved.

The memory can be dynamically switched from one configuration to another by changing the MS bit in OMR. The address ranges that are directly affected by the switch operation are specified in Table 3-3. The memory switch can be accomplished provided that the affected address ranges are not being accessed during the instruction cycle in which the switch operation takes place. For trouble-free dynamic switching, no accesses (including instruction fetches) to or from the affected address ranges in program and data memories are allowed during the switch cycle.

NOTE

The switch cycle actually occurs three instruction cycles after the instruction that modifies the MS bit.

Any sequence that complies with the switch condition is valid. For example, if the program flow executes in the address range that is not affected by the switch, the switch condition can be met very easily. In this case a switch can be accomplished by just changing the MS bit in OMR in the regular program flow, assuming no accesses to the affected address ranges of the data memory occur up to three instructions after the instruction that changes the OMR bit. Special care should be taken in relation to the interrupt vector routines since an interrupt could cause the DSP to fetch instructions out of sequence and might violate the switch condition.

Special attention should be given when running a memory switch routine using the OnCE port. Running the switch routine in trace mode, for example, can cause the switch to complete after the MS bit change while the DSP is in debug mode. As a result, subsequent instructions might be fetched according to the new memory configuration (after the switch), and thus might execute improperly.

3.4 Memory Maps

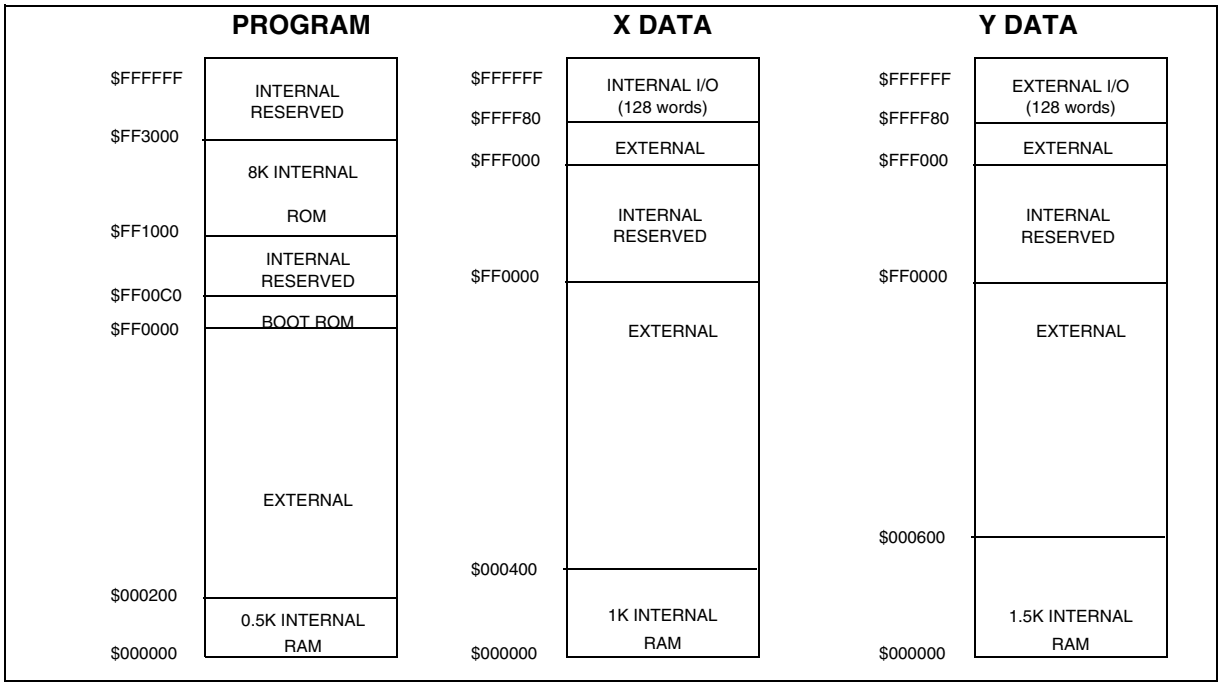


Figure 3-1 Memory Maps for MS=0, SC=0

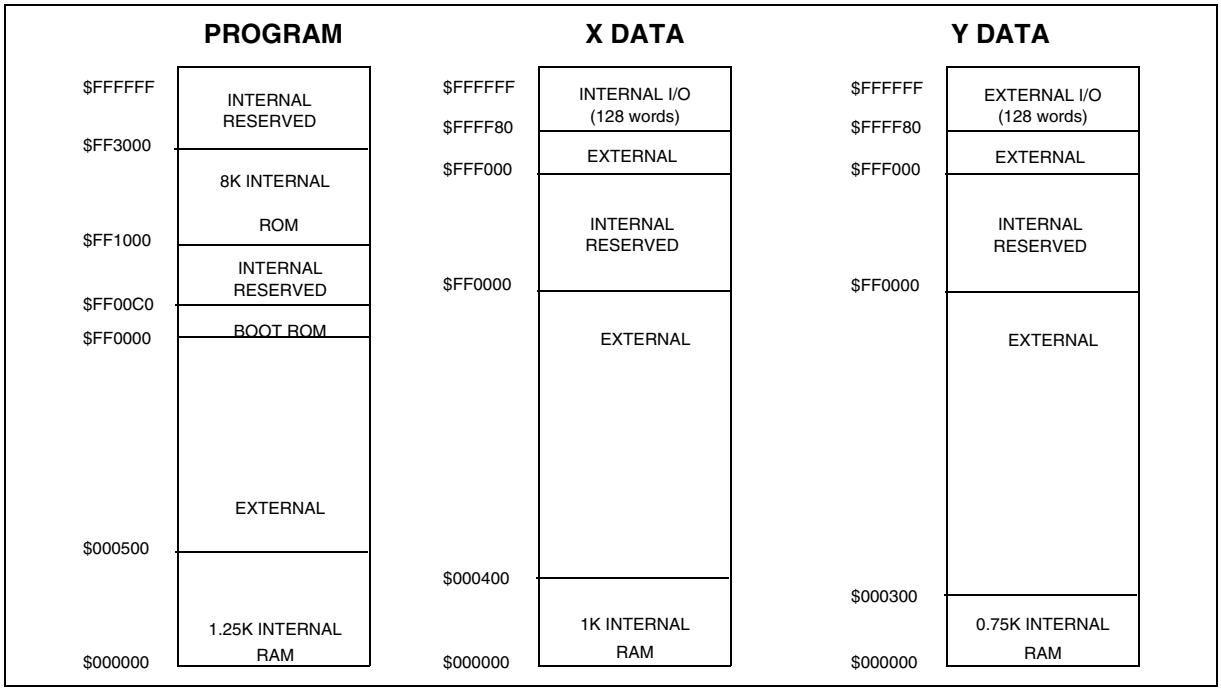


Figure 3-2 Memory Maps for MS=1, SC=0

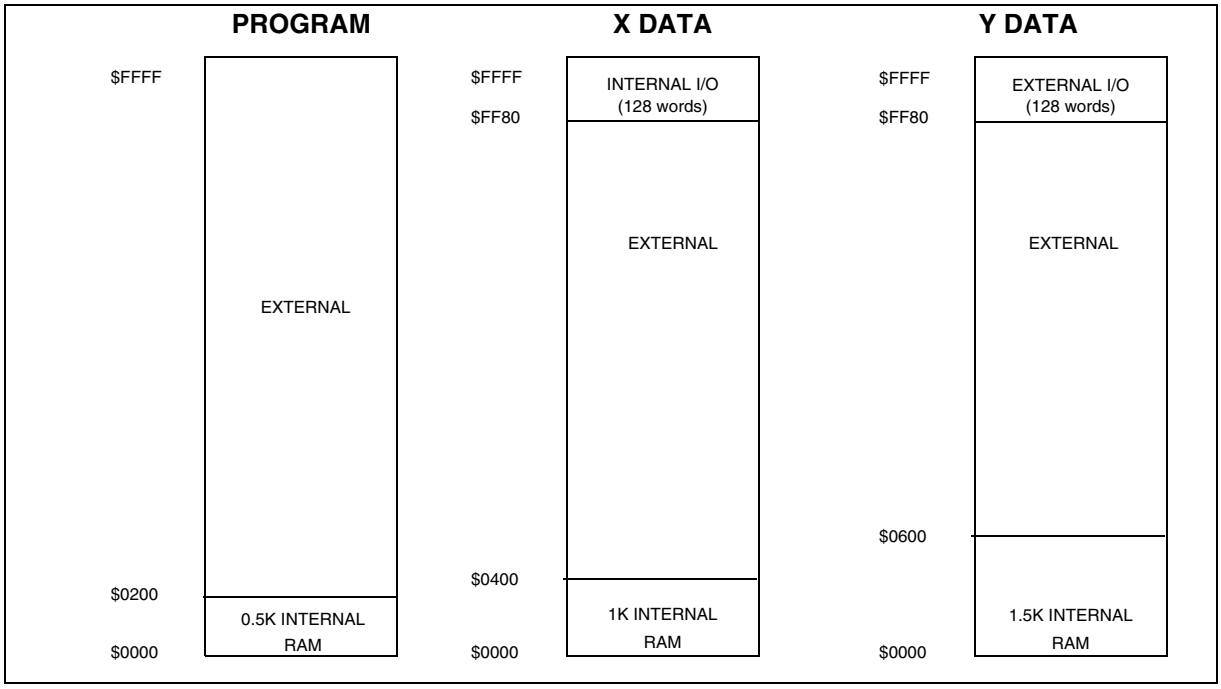


Figure 3-3 Memory Maps for MS=0, SC=1

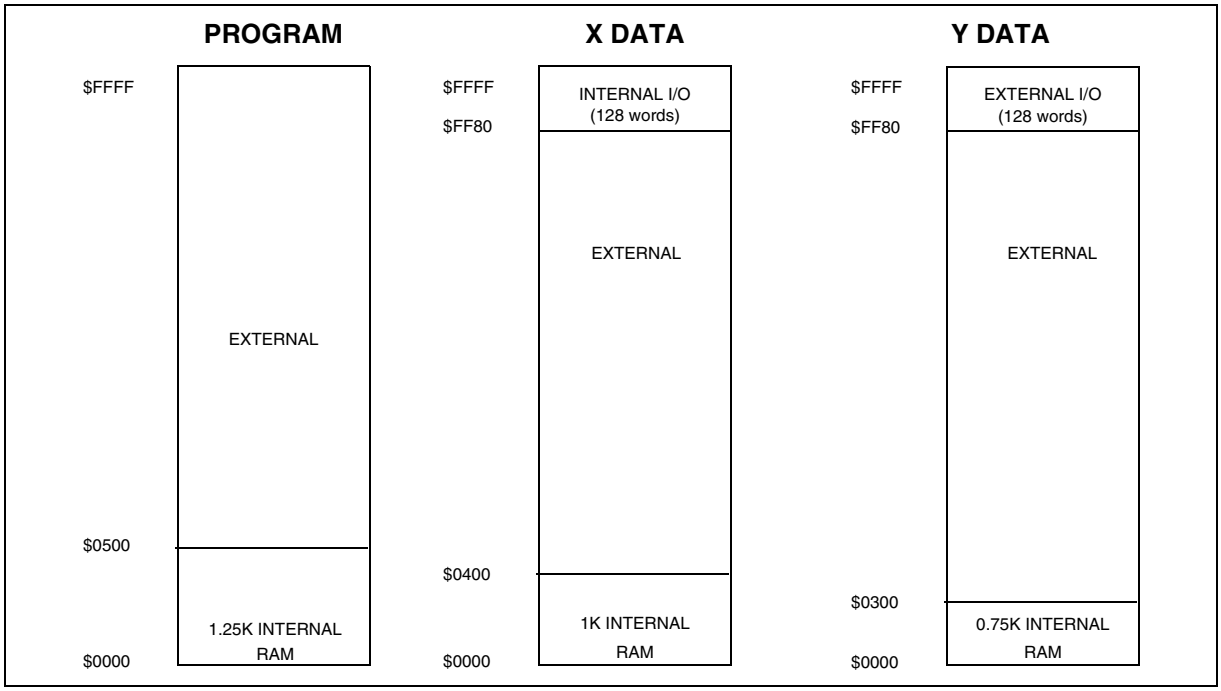


Figure 3-4 Memory Maps for MS=1, SC=1

3.5 External Memory Support

The DSP56364 does not support the SSRAM memory type. It does support SRAM and DRAM as indicated in the DSP56300 24-Bit Digital Signal Processor Family Manual, Freescale publication DSP56300FM/AD. Note that the DSP56364 has only an 8-bit data bus. This means that no instruction fetches from external memory are possible, and care should be taken to ensure that no program memory instruction fetch access occurs in the external memory space. The DMA may be used to automatically pack and unpack 24-bit data into the 8-bit wide external memory, during DMA data transfers.

Also, care should be taken when accessing external memory to ensure that the necessary address lines are available. For example, when using glueless SRAM interfacing, it is possible to directly address 2^{19} memory locations (512K) when using the 18 address lines and the two programmable address attribute lines. Using DRAM access mode, the full 16M addressing range may be used.

3.6 Internal I/O Memory Map

The DSP56364 on-chip peripheral modules have their register files programmed to the addresses in the internal X-I/O memory range (the top 128 locations of the X data memory space) as shown in Appendix A, "Bootstrap ROM".

4 Core Configuration

4.1 Introduction

This chapter contains DSP56300 core configuration information details specific to the DSP56364. These include the following:

- Operating modes
- Bootstrap program
- Interrupt sources and priorities
- DMA request sources
- OMR
- PLL control register
- AA control registers
- JTAG BSR

For more information on specific registers or modules in the DSP56300 core, refer to the *DSP56300 Family Manual (DSP56300FM)*.

4.2 Operating Mode Register (OMR)

The contents of the Operating Mode Register (OMR) are shown in Figure 4-1. Refer to the *DSP56300 24-Bit Digital Signal Processor Family Manual*, Freescale publication DSP56300FM for a description of the OMR bits.

SCS								EOM								COM							
23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
			SEN	WRP	EOV	EUN	XYS	ATE	APD			TAS		CDP1:0		MS	SD		EBD	MD	MC	MB	MA

		ATE	- Address Tracing Enable	MS	- Memory Switch Mode
		APD	- Address Priority Disable	SD	- Stop Delay
SEN	- Stack Extension Enable			EBD	- External Bus Disable
WRP	- Extended Stack Wrap Flag	TAS	- TA Synchronize Select	MD	- Operating Mode D
EOV	- Extended Stack Overflow Flag			MC	- Operating Mode C - Always set.
EUN	- Extended Stack Underflow Flag	CDP1	- Core-Dma Priority 1	MB	- Operating Mode B
XYS	- Stack Extension Space Select	CDP0	- Core-Dma Priority 0	MA	- Operating Mode A
	- Reserved bit. Read as zero, should be written with zero for future compatibility				

Figure 4-1 Operating Mode Register (OMR)

4.2.1 Mode C (MC) - Bit 2

The Mode C (MC) bit is set during hardware reset and should be left set in the DSP56364.

4.2.2 Address Attribute Priority Disable (APD) - Bit 14

The Address Attribute Priority Disable (APD) bit is used to turn off the address attribute priority mechanism. When this bit is set, more than one address attribute pin AA/$\overline{\text{RAS}}$(1:0) may be simultaneously asserted according to its AAR settings. The APD bit is cleared by hardware reset.

4.2.3 Address Tracing Enable (ATE) - Bit 15

The Address Tracing Enable (ATE) bit is used to turn on Address Tracing (AT) Mode. When the AT Mode is enabled, the DSP56300 Core reflects the addresses of internal fetches and program space moves (MOVEM) to the Address Bus (A0-A17), if the Address Bus is not needed by the DSP56300 Core for external accesses. The ATE bit is cleared on hardware reset.

4.3 Operating Modes

The operating modes are as shown in Table 4-1 The operating modes are latched from MODA, MODB and MODD pins during reset. Each operating mode is briefly described below. The operation of all bootstrap modes is defined by the Bootstrap ROM source code in Appendix A, "Bootstrap ROM".

Table 4-1 DSP56364 Operating Modes

Mode	MOD D	MOD B	MOD A	Reset Vector	Description
$4	0	0	0	$FF0000	Jump to PROM starting address
$5	0	0	1	$FF0000	Bootstrap from byte-wide memory
$6	0	1	0	$FF0000	Reserved
$7	0	1	1	$FF0000	Reserved for Burn-in testing
$C	1	0	0	$FF0000	Reserved
$D	1	0	1	$FF0000	Bootstrap from SHI (slave SPI mode)
$E	1	1	0	$FF0000	Bootstrap from SHI (slave I^2C mode, clock freeze enabled)
$F	1	1	1	$FF0000	Bootstrap from SHI (slave I^2C mode, clock freeze disabled)

Mode 4 The DSP starts fetching instructions from the starting address of the on-chip Program ROM.

Mode 5 The bootstrap program loads instructions through Port A from external byte-wide memory starting at address P:$D00000. The bootstrap code expects to read 3 bytes specifying the number of program words, 3 bytes specifying the address to start loading the program words and then 3 bytes for each program word to be loaded. The number of words, the starting address and the program words are read least significant byte first followed by the mid and then by the most significant byte. The program words will be stored in contiguous PRAM memory locations starting at the specified starting address. After reading the program words, program execution starts from the same address where loading started. The SRAM memory access type is selected by the values in Address Attribute Register 1 (AAR1), with 31 wait states for each memory access. Address $D00000 is reflected as address $00000 on Port A pins A0-A17.

Mode 6 Reserved.

Mode 7 Reserved for Burn-In testing.

Mode C Reserved.

Mode D In this mode, the internal PRAM is loaded from the Serial Host Interface (SHI). The SHI operates in the SPI slave mode, with 24-bit word width.The bootstrap code expects to read a 24-bit word specifying the number of program words, a 24-bit word specifying the

address to start loading the program words and then a 24-bit word for each program word to be loaded. The program words will be stored in contiguous PRAM memory locations starting at the specified starting address. After reading the program words, program execution starts from the same address where loading started.

Mode E — Same as mode 5, except the SHI interface operates in the I^2C slave mode with clock freeze enabled.

Mode F — Same as mode 5, except the SHI interface operates in the I^2C slave mode with clock freeze disabled (compatible to DSP56000 family).

4.4 Bootstrap Program

The bootstrap program is factory-programmed in an internal 192 word by 24-bit bootstrap ROM located in program memory space at locations $FF0000–$FF00BF. The bootstrap program can load any program RAM segment from an external byte-wide EPROM or the SHI. The bootstrap program described here, and listed in **Appendix A**, is a default, which may be modified or replaced by the customer.

On exiting the Reset state, the DSP56364 does the following:

1. Samples the MODA, MODB, and MODD signal lines
2. Loads their values into bits MA, MB, and MD in the OMR

The contents of the MA, MB, MC, and MD bits determine which bootstrap mode the DSP56364 enters.

See Table 4-1 for a tabular description of the mode bit settings for the operating modes.

The bootstrap program options can be invoked at any time by setting the appropriate MA, MB, and MD bits in the OMR and jumping to the bootstrap program entry point, $FF0000. The mode selection bits in the OMR can be set directly by software.It is recommended to keep the MC bit set to 1.

In bootstrap modes 5, D, E, and F, the bootstrap program expects the following data sequence when downloading the user program through an external port:

1. Three bytes defining the number of (24-bit) program words to be loaded
2. Three bytes defining the (24-bit) start address to which the user program loads in the DSP56364 program memory
3. The user program (three bytes for each 24-bit program word). The program words will be stored in contiguous PRAM memory locations starting at the specified starting address.

The three bytes for each data sequence must be loaded with the least significant byte first.

Once the bootstrap program completes loading the specified number of words, it jumps to the specified starting address and executes the loaded program.

4.5 Interrupt Priority Registers

There are two interrupt priority registers in the DSP56364: IPR-C is dedicated for DSP56300 Core interrupt sources and IPR-P is dedicated for DSP56364 peripheral interrupt sources. The interrupt priority registers are shown in Figure 4-2 and Figure 4-3 The Interrupt Priority Level bits are defined in Table 4-2.

The interrupt vectors are shown in Table C-2 and the interrupt priorities are shown in Table C-3 in Appendix C, "Programmer's Reference".

Table 4-2 Interrupt Priority Level Bits

IPL bits		Interrupts Enabled	Interrupt Priority Level
xxL1	xxL0		
0	0	No	—
0	1	Yes	0
1	0	Yes	1
1	1	Yes	2

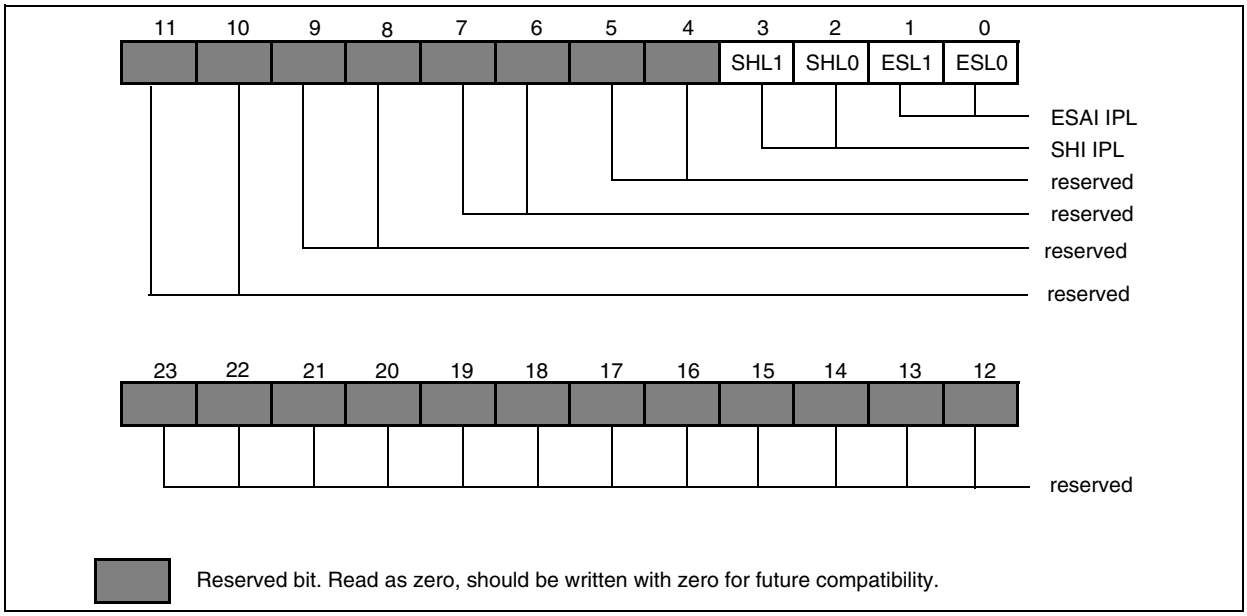


Figure 4-2 Interrupt Priority Register P

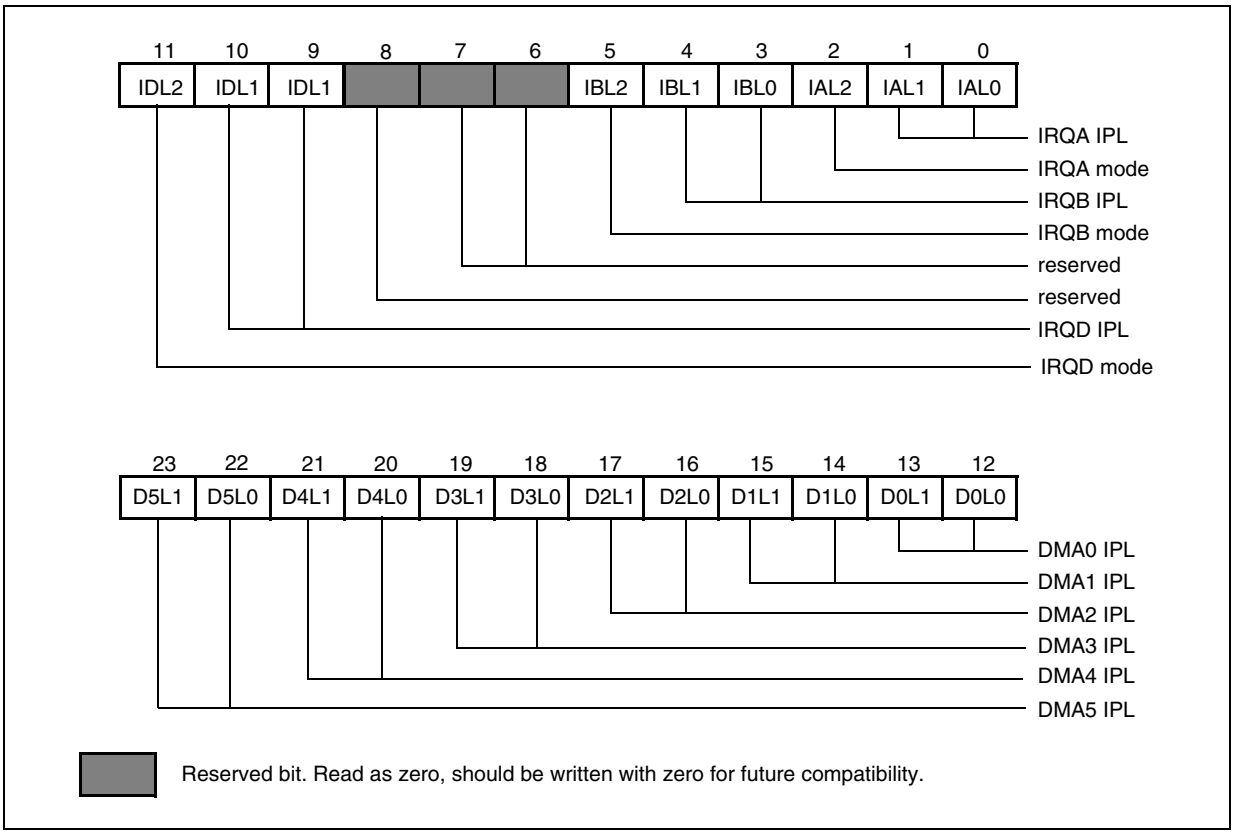


Figure 4-3 Interrupt Priority Register C

4.6 DMA Request Sources

The DMA Request Source bits (DRS0-DRS4 bits in the DMA Control/Status registers) encode the source of DMA requests used to trigger the DMA transfers. The DMA request sources may be the internal peripherals or external devices requesting service through the $\overline{\text{IRQA}}$, $\overline{\text{IRQB}}$ and $\overline{\text{IRQD}}$ pins. The DMA Request Sources are shown in Table 4-3.

Table 4-3 DMA Request Sources

DMA Request Source Bits DRS4...DRS0	Requesting Device
00000	External ($\overline{\text{IRQA}}$ pin)
00001	External ($\overline{\text{IRQB}}$ pin)
00010	Reserved
00011	External ($\overline{\text{IRQD}}$ pin)
00100	Transfer Done from DMA channel 0
00101	Transfer Done from DMA channel 1
00110	Transfer Done from DMA channel 2
00111	Transfer Done from DMA channel 3
01000	Transfer Done from DMA channel 4
01001	Transfer Done from DMA channel 5
01010	Reserved
01011	ESAI Receive Data (RDF=1)
01100	ESAI Transmit Data (TDE=1)
01101	SHI HTX Empty
01110	SHI FIFO Not Empty
01111	SHI FIFO Full
10000-11111	RESERVED

4.7 PLL and Clock Generator

4.7.1 PLL Multiplication Factor (MF0-MF11) - Bits 0-11

The DSP56364 PLL multiplication factor is set to 6 during hardware reset, i.e. the Multiplication Factor Bits MF0-MF11 in the PLL Control Register (PCTL) are set to $005.

4.7.2 Crystal Range Bit (XTLR) - Bit 15

The Crystal Range (XTLR) bit controls the on-chip crystal oscillator transconductance. The on-chip crystal oscillator is not used on the DSP56364 since no XTAL pin is available. The XTLR bit is set to zero during hardware reset in the DSP56364.

4.7.3 XTAL Disable Bit (XTLD) - Bit 16

The XTAL Disable Bit (XTLD) is set to 1 (XTAL disabled) during hardware reset in the DSP56364.

4.7.4 Clock Output Disable Bit (COD) - Bit 19

The Clock Output Disable Bit (COD) is set to 0 during hardware reset. Since no clock output pin is available in the DSP56364, this bit does not affect the functionality of the clock generator.

4.7.5 PLL Pre-Divider Factor (PD0-PD3) - Bits 20-23

The DSP56364 PLL Pre-Divider factor is set to 1 during hardware reset, i.e. the Pre-Divider Factor Bits PD0-PD3 in the PLL Control Register (PCTL) are set to $0.

4.8 Device Identification (ID) Register

The Device Identification Register (IDR) is a 24 bit read only factory programmed register used to identify the different DSP56300 core-based family members. This register specifies the derivative number and revision number. This information may be used in testing or by software. Table 4-4 shows the ID register configuration.

Table 4-4 Identification Register Configuration

23 16	15 12	11 0
Reserved	Revision Number	Derivative Number
$00	$0	$364

4.9 JTAG Identification (ID) Register

The JTAG Identification (ID) Register is a 32 bit, read only thought JTAG, factory programmed register used to distinguish the component on a board according to the IEEE 1149.1 standard. Table 4-5 shows the JTAG ID register configuration.

Table 4-5 JTAG Identification Register Configuration

31 28	27 22	21 12	11 1	0
Version Information	**Customer Part Number**	**Sequence Number**	**Manufacturer Identity**	**1**
0000	000110	0001100100	00000001110	1

4.10 JTAG Boundary Scan Register (BSR)

The boundary scan register (BSR) in the DSP56364 JTAG implementation contains bits for all device signal and clock pins and associated control signals. All bidirectional pins have a single register bit in the boundary scan register for pin data, and are controlled by an associated control bit in the boundary scan register. The boundary scan register bit definitions are described in Table 4-6 The BSDL file may be found in Appendix B, "BDSL File".

Table 4-6 DSP56364 BSR Bit Definition

Bit #	Pin Name	Pin Type	BSR Cell Type
	GPIO3	-	Control
	GPIO3	Input/Output	Data
	GPIO2	-	Control
	GPIO2	Input/Output	Data
	GPIO1	-	Control
	GPIO1	Input/Output	Data
	GPIO0	-	Control
	GPIO0	Input/Output	Data
	D7	Input/Output	Data
	D6	Input/Output	Data
	D5	Input/Output	Data
	D4	Input/Output	Data
	D[7:0]	-	Control
	D3	Input/Output	Data
	D2	Input/Output	Data
	D1	Input/Output	Data
	D0	Input/Output	Data
	A17	Output3	Data
	A16	Output3	Data
	A15	Output3	Data
	A14	Output3	Data
	A13	Output3	Data

Bit #	Pin Name	Pin Type	BSR Cell Type
	WR	Output3	Data
	CAS	-	Control
	CAS	Output3	Data
	TA	Input	Data
	EXTAL	Input	Data
	RES	Input	Data
	PINIT	Input	Data
	HREQ	-	Control
	HREQ	Input/Output	Data
	SCK/SCL	-	Control
	SCK/SCL	Input/Output	Data
	MISO/SDA	-	Control
	MISO/SDA	Input/Output	Data
	MOSI/HA0	-	Control
	MOSI/HA0	Input/Output	Data
	SS	Input	Data
	SDO5/SDI0	-	Control
	SDO5/SDI0	Input/Output	Data
	SDO4/SDI1	-	Control
	SDO4/SDI1	Input/Output	Data
	SDO3/SDI2	-	Control
	SDO3/SDI2	Input/Output	Data

Table 4-6 DSP56364 BSR Bit Definition (continued)

Bit #	Pin Name	Pin Type	BSR Cell Type
	A12	Output3	Data
	A[17:9]	-	Control
	A11	Output3	Data
	A10	Output3	Data
	A9	Output3	Data
	A8	Output3	Data
	A7	Output3	Data
	A[8:0]	-	Control
	A6	Output3	Data
	A5	Output3	Data
	A4	Output3	Data
	A3	Output3	Data
	A2	Output3	Data
	A1	Output3	Data
	A0	Output3	Data
	AA0	-	Control
	AA0	Output3	Data
	AA1	-	Control
	AA1	Output3	Data
	$\overline{RD}$,$\overline{WR}$	-	Control
	RD	Output3	Data

Bit #	Pin Name	Pin Type	BSR Cell Type
	SDO2/SDI3	-	Control
	SDO2/SDI3	Input/Output	Data
	SDO1	-	Control
	SDO1	Input/Output	Data
	SDO0	-	Control
	SDO0	Input/Output	Data
	HCKR	-	Control
	HCKR	Input/Output	Data
	HCKT	-	Control
	HCKT	Input/Output	Data
	SCKR	-	Control
	SCKR	Input/Output	Data
	SCKT	-	Control
	SCKT	Input/Output	Data
	FSR	-	Control
	FSR	Input/Output	Data
	FST	-	Control
	FST	Input/Output	Data
	MODA	Input	Data
	MODB	Input	Data
	MODD	Input	Data

5 General Purpose Input/Output Port (GPIO)

5.1 Introduction

The General Purpose Input/Output (GPIO) pins are used for control and handshake functions between the DSP and external circuitry. The GPIO port has 4 I/O pins (GPIO0-GPIO3) that are controlled through a set of memory-mapped registers. Each GPIO pin may be individually programmed as an output or as an input.

5.2 GPIO Programming Model

The GPIO port is controlled by three registers: Port B Control register (PCRB), Port B Direction register (PRRB) and Port B GPIO Data register (PDRB). These registers are shown in Figure 5-1, Figure 5-2, and Figure 5-3.

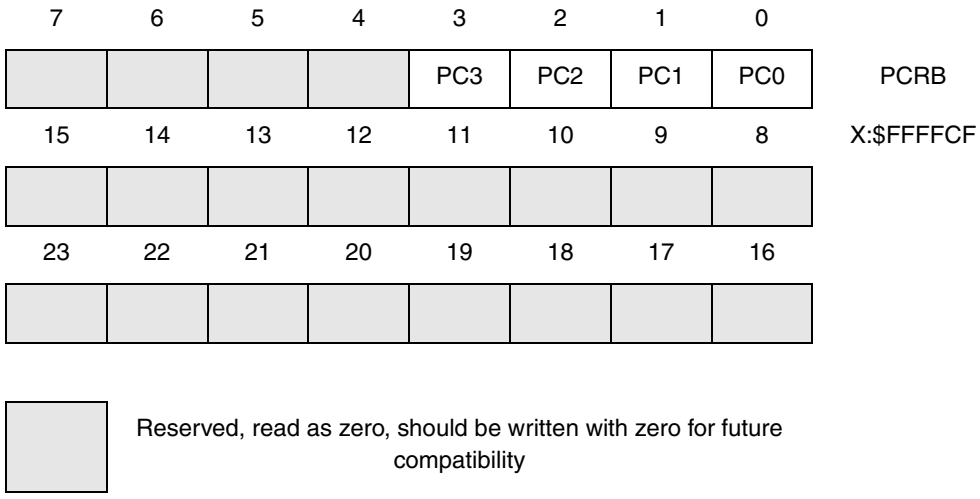


Figure 5-1 GPIO Port B Control Register (PCRB)

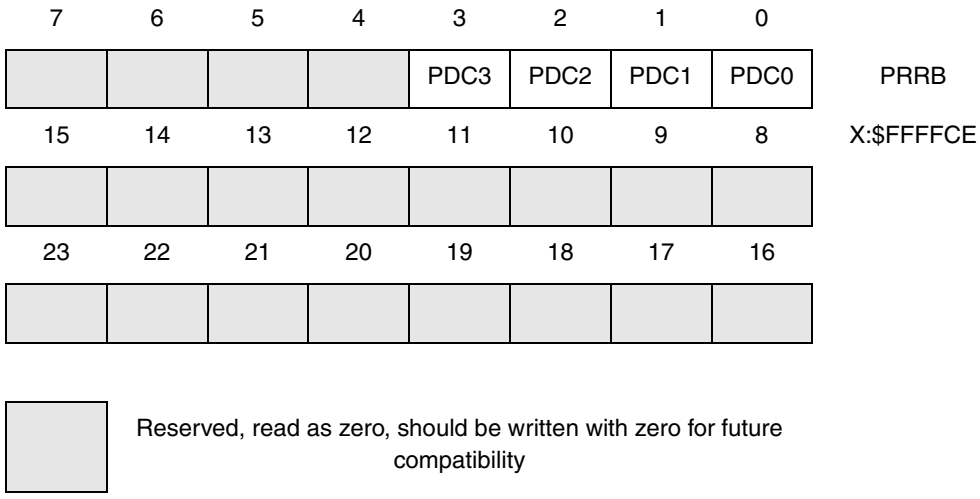


Figure 5-2 GPIO Port B Direction Register (PRRB)

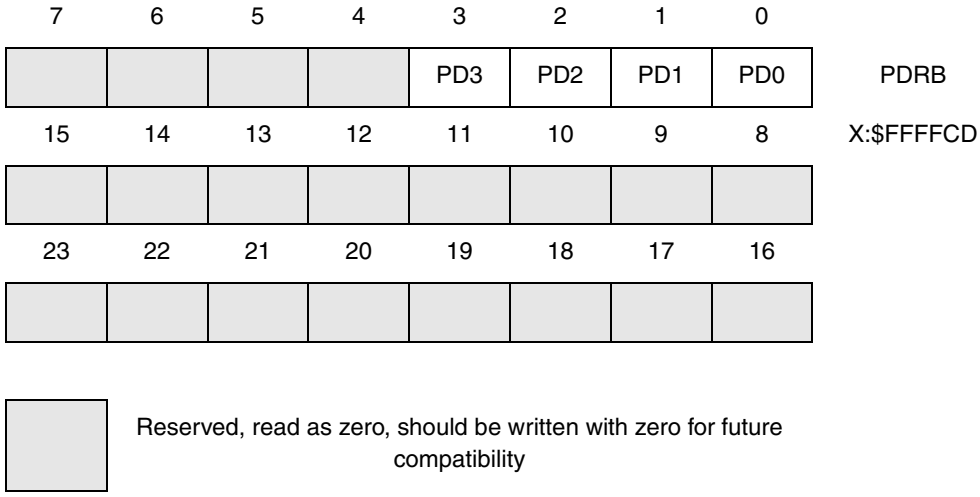


Figure 5-3 GPIO Port B Data Register (PDRB)

5.2.1 Port B Control Register (PCRB)

The read/write Port B Control Register (PCRB) controls the functionality of the GPIO pins in conjunction with the Port B Direction Register (PRRB).

5.2.1.1 PCRB Control Bits (PC[3:0]) - Bits 3-0

When a PC[i] bit is cleared, the corresponding GPIO[i] pin is three-stated if PDC[i] is cleared, or it is an output if PDC[i] is set. When a PC[i] bit is set, the corresponding GPIO[i] pin is an input if PDC[i] is cleared, or it is an open-drain output if PDC[i] is set. Refer to Table 5-1 for a summary of the GPIO configuration control. Hardware and software reset clear the PC[3:0] bits.

5.2.1.2 PCRB Reserved Bits - Bits 23-4

These bits are reserved and unused. They read as 0s and should be written with 0s for future compatibility.

Port B Direction Register (PRRB)

The read/write Port B Direction Register controls the direction of data transfer for each GPIO pin.

5.2.1.3 PRRB Direction Bits (PDC[3:0]) - Bits 3-0

When PDC[i] is set, the GPIO port pin[i] is configured as output. When PDC[i] is cleared the GPIO port pin[i] is configured as input. See Table 5-1. Hardware and software reset clear the PDC[3:0] bits.

5.2.1.4 PRRB Reserved Bits - Bits 23-4

These bits are reserved and unused. They read as 0s and should be written with 0s for future compatibility.

Table 5-1 GPIO Pin Configuration

PDC[i]	PC[i]	GPIO Pin[i] Function
0	0	Three-Stated (Disconnected)
0	1	Input
1	0	Output
1	1	Open-drain output

5.2.2 Port B GPIO Data Register (PDRB)

The read/write Port B Data Register (PDRB) is used to read data from or write data to the GPIO pins.

5.2.2.1 PDRB Data Bits (PD[3:0]) - Bits 3-0

If a GPIO pin [i] is configured as a GPIO input, then the corresponding PD[i] bit will reflect the value present on this pin. If a GPIO pin [i] is configured as a GPIO output, then the value written into the corresponding PD[i] bit will be reflected on the pin. The PD[3:0] bits are not affected by hardware or software reset.

5.2.2.2 PDRB Reserved Bits - Bits 23-4

These bits are reserved and unused. They read as 0s and should be written with 0s for future compatibility.

NOTES

6 Enhanced Serial AUDIO Interface (ESAI)

6.1 Introduction

The Enhanced Serial Audio Interface (ESAI) provides a full-duplex serial port for serial communication with a variety of serial devices including one or more industry-standard codecs, other DSPs, microprocessors, and peripherals which implement the Freescale SPI serial protocol. The ESAI consists of independent transmitter and receiver sections, each section with its own clock generator. It is a superset of the 56300 Family ESSI peripheral and of the 56000 Family SAI peripheral.

The ESAI block diagram is shown in Figure 6-1. The ESAI is named synchronous because all serial transfers are synchronized to a clock. Additional synchronization signals are used to delineate the word frames. The normal mode of operation is used to transfer data at a periodic rate, one word per period. The network mode is similar in that it is also intended for periodic transfers; however, it supports up to 32 words (time slots) per period. This mode can be used to build time division multiplexed (TDM) networks. In contrast, the on-demand mode is intended for non-periodic transfers of data and to transfer data serially at high speed when the data becomes available. This mode offers a subset of the SPI protocol.

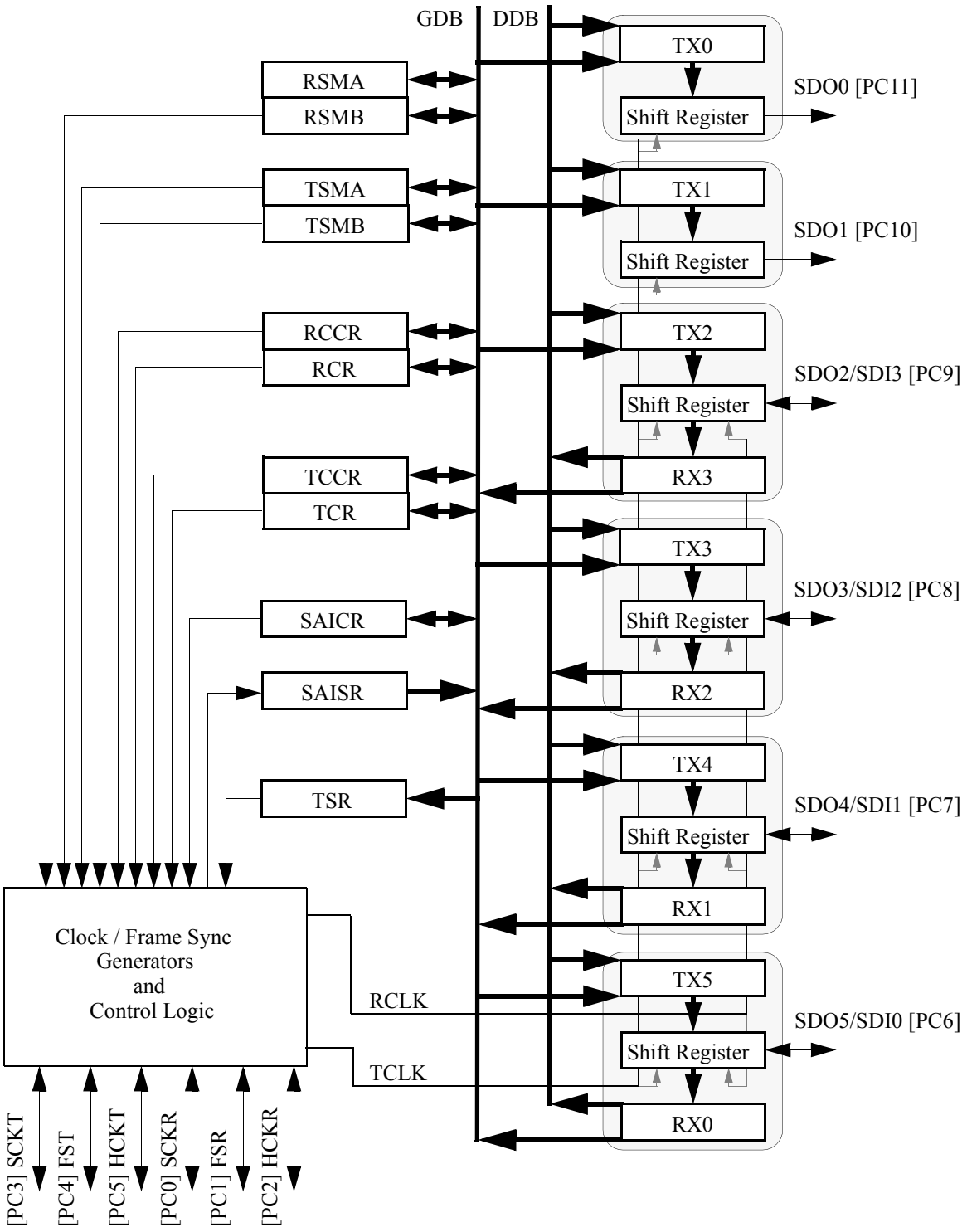


Figure 6-1 ESAI Block Diagram

6.2 ESAI Data and Control Pins

Three to twelve pins are required for operation, depending on the operating mode selected and the number of transmitters and receivers enabled. The SDO0 and SDO1 pins are used by transmitters 0 and 1 only. The SDO2/SDI3, SDO3/SDI2, SDO4/SDI1, and SDO5/SDI0 pins are shared by transmitters 2 to 5 with receivers 0 to 3. The actual mode of operation is selected under software control. All transmitters operate fully synchronized under control of the same transmitter clock signals. All receivers operate fully synchronized under control of the same receiver clock signals.

6.2.1 Serial Transmit 0 Data Pin (SDO0)

SDO0 is used for transmitting data from the TX0 serial transmit shift register. SDO0 is an output when data is being transmitted from the TX0 shift register. In the on-demand mode with an internally generated bit clock, the SDO0 pin becomes high impedance for a full clock period after the last data bit has been transmitted, assuming another data word does not follow immediately. If a data word follows immediately, there is no high-impedance interval.

SDO0 may be programmed as a general-purpose I/O pin (PC11) when the ESAI SDO0 function is not being used.

6.2.2 Serial Transmit 1 Data Pin (SDO1)

SDO1 is used for transmitting data from the TX1 serial transmit shift register. SDO1 is an output when data is being transmitted from the TX1 shift register. In the on-demand mode with an internally generated bit clock, the SDO1 pin becomes high impedance for a full clock period after the last data bit has been transmitted, assuming another data word does not follow immediately. If a data word follows immediately, there is no high-impedance interval.

SDO1 may be programmed as a general-purpose I/O pin (PC10) when the ESAI SDO1 function is not being used.

6.2.3 Serial Transmit 2/Receive 3 Data Pin (SDO2/SDI3)

SDO2/SDI3 is used as the SDO2 for transmitting data from the TX2 serial transmit shift register when programmed as a transmitter pin, or as the SDI3 signal for receiving serial data to the RX3 serial receive shift register when programmed as a receiver pin. SDO2/SDI3 is an input when data is being received by the RX3 shift register. SDO2/SDI3 is an output when data is being transmitted from the TX2 shift register. In the on-demand mode with an internally generated bit clock, the SDO2/SDI3 pin becomes high impedance for a full clock period after the last data bit has been transmitted, assuming another data word does not follow immediately. If a data word follows immediately, there is no high-impedance interval.

SDO2/SDI3 may be programmed as a general-purpose I/O pin (PC9) when the ESAI SDO2 and SDI3 functions are not being used.

6.2.4 Serial Transmit 3/Receive 2 Data Pin (SDO3/SDI2)

SDO3/SDI2 is used as the SDO3 signal for transmitting data from the TX3 serial transmit shift register when programmed as a transmitter pin, or as the SDI2 signal for receiving serial data to the RX2 serial receive shift register when programmed as a receiver pin. SDO3/SDI2 is an input when data is being received by the RX2 shift register. SDO3/SDI2 is an output when data is being transmitted from the TX3 shift register. In the on-demand mode with an internally generated bit clock, the SDO3/SDI2 pin becomes high impedance for a full clock period after the last data bit has been transmitted, assuming another data word does not follow immediately. If a data word follows immediately, there is no high-impedance interval.

SDO3/SDI2 may be programmed as a general-purpose I/O pin (PC8) when the ESAI SDO3 and SDI2 functions are not being used.

6.2.5 Serial Transmit 4/Receive 1 Data Pin (SDO4/SDI1)

SDO4/SDI1 is used as the SDO4 signal for transmitting data from the TX4 serial transmit shift register when programmed as transmitter pin, or as the SDI1 signal for receiving serial data to the RX1 serial receive shift register when programmed as a receiver pin. SDO4/SDI1 is an input when data is being received by the RX1 shift register. SDO4/SDI1 is an output when data is being transmitted from the TX4 shift register. In the on-demand mode with an internally generated bit clock, the SDO4/SDI1 pin becomes high impedance for a full clock period after the last data bit has been transmitted, assuming another data word does not follow immediately. If a data word follows immediately, there is no high-impedance interval.

SDO4/SDI1 may be programmed as a general-purpose I/O pin (PC7) when the ESAI SDO4 and SDI1 functions are not being used.

6.2.6 Serial Transmit 5/Receive 0 Data Pin (SDO5/SDI0)

SDO5/SDI0 is used as the SDO5 signal for transmitting data from the TX5 serial transmit shift register when programmed as transmitter pin, or as the SDI0 signal for receiving serial data to the RX0 serial shift register when programmed as a receiver pin. SDO5/SDI0 is an input when data is being received by the RX0 shift register. SDO5/SDI0 is an output when data is being transmitted from the TX5 shift register. In the on-demand mode with an internally generated bit clock, the SDO5/SDI0 pin becomes high impedance for a full clock period after the last data bit has been transmitted, assuming another data word does not follow immediately. If a data word follows immediately, there is no high-impedance interval.

SDO5/SDI0 may be programmed as a general-purpose I/O pin (PC6) when the ESAI SDO5 and SDI0 functions are not being used

6.2.7 Receiver Serial Clock (SCKR)

SCKR is a bidirectional pin providing the receivers serial bit clock for the ESAI interface. The direction of this pin is determined by the RCKD bit in the RCCR register.The SCKR operates as a clock input or output used by all the enabled receivers in the asynchronous mode (SYN=0), or as serial flag 0 pin in the synchronous mode (SYN=1).

When this pin is configured as serial flag pin, its direction is determined by the RCKD bit in the RCCR register. When configured as the output flag OF0, this pin reflects the value of the OF0 bit in the SAICR register, and the data in the OF0 bit shows up at the pin synchronized to the frame sync being used by the transmitter and receiver sections. When this pin is configured as the input flag IF0, the data value at the pin is stored in the IF0 bit in the SAISR register, synchronized by the frame sync in normal mode or the slot in network mode.

SCKR may be programmed as a general-purpose I/O pin (PC0) when the ESAI SCKR function is not being used.

NOTE

Although the external ESAI serial clock can be independent of and asynchronous to the DSP system clock, the DSP clock frequency must be at least three times the external ESAI serial clock frequency and each ESAI serial clock phase must exceed the minimum of 1.5 DSP clock periods.

For more information on pin mode and definition, see Table 6-7 and on receiver clock signals see Table 6-1.

Table 6-1 Receiver Clock Sources (asynchronous mode only)

RHCKD	RFSD	RCKD	Receiver Bit Clock Source	OUTPUTS		
0	0	0	SCKR			
0	0	1	HCKR			SCKR
0	1	0	SCKR		FSR	
0	1	1	HCKR		FSR	SCKR
1	0	0	SCKR	HCKR		
1	0	1	INT	HCKR		SCKR
1	1	0	SCKR	HCKR	FSR	
1	1	1	INT	HCKR	FSR	SCKR

6.2.8 Transmitter Serial Clock (SCKT)

SCKT is a bidirectional pin providing the transmitters serial bit clock for the ESAI interface. The direction of this pin is determined by the TCKD bit in the TCCR register. The SCKT is a clock input or output used by all the enabled transmitters in the asynchronous mode (SYN=0) or by all the enabled transmitters and receivers in the synchronous mode (SYN=1) (see Table 6-2).

Table 6-2 Transmitter Clock Sources

THCKD	TFSD	TCKD	Transmitter Bit Clock Source	OUTPUTS		
0	0	0	SCKT			
0	0	1	HCKT			SCKT
0	1	0	SCKT		FST	
0	1	1	HCKT		FST	SCKT
1	0	0	SCKT	HCKT		
1	0	1	INT	HCKT		SCKT
1	1	0	SCKT	HCKT	FST	
1	1	1	INT	HCKT	FST	SCKT

SCKT may be programmed as a general-purpose I/O pin (PC3) when the ESAI SCKT function is not being used.

NOTE

Although the external ESAI serial clock can be independent of and asynchronous to the DSP system clock, the DSP clock frequency must be at least three times the external ESAI serial clock frequency and each ESAI serial clock phase must exceed the minimum of 1.5 DSP clock periods.

6.2.9 Frame Sync for Receiver (FSR)

FSR is a bidirectional pin providing the receivers frame sync signal for the ESAI interface. The direction of this pin is determined by the RFSD bit in RCR register. In the asynchronous mode (SYN=0), the FSR pin operates as the frame sync input or output used by all the enabled receivers. In the synchronous mode (SYN=1), it operates as either the serial flag 1 pin (TEBE=0), or as the transmitter external buffer enable control (TEBE=1, RFSD=1). For further information on pin mode and definition, see Table 6-8 and on receiver clock signals see Table 6-1.

When this pin is configured as serial flag pin, its direction is determined by the RFSD bit in the RCCR register. When configured as the output flag OF1, this pin reflects the value of the OF1 bit in the SAICR register, and the data in the OF1 bit shows up at the pin synchronized to the frame sync being used by the transmitter and receiver sections. When configured as the input flag IF1, the data value at the pin is stored in the IF1 bit in the SAISR register, synchronized by the frame sync in normal mode or the slot in network mode.

FSR may be programmed as a general-purpose I/O pin (PC1) when the ESAI FSR function is not being used.

6.2.10 Frame Sync for Transmitter (FST)

FST is a bidirectional pin providing the frame sync for both the transmitters and receivers in the synchronous mode (SYN=1) and for the transmitters only in asynchronous mode (SYN=0) (see Table 6-2). The direction of this pin is determined by the TFSD bit in the TCR register. When configured as an output, this pin is the internally generated frame sync signal. When configured as an input, this pin receives an external frame sync signal for the transmitters (and the receivers in synchronous mode).

FST may be programmed as a general-purpose I/O pin (PC4) when the ESAI FST function is not being used.

6.2.11 High Frequency Clock for Transmitter (HCKT)

HCKT is a bidirectional pin providing the transmitters high frequency clock for the ESAI interface. The direction of this pin is determined by the THCKD bit in the TCCR register. In the asynchronous mode (SYN=0), the HCKT pin operates as the high frequency clock input or output used by all enabled transmitters. In the synchronous mode (SYN=1), it operates as the high frequency clock input or output used by all enabled transmitters and receivers. When programmed as input this pin is used as an alternative high frequency clock source to the ESAI transmitter rather than the DSP main clock. When programmed as output it can serve as a high frequency sample clock (to external DACs for example) or as an additional system clock. See Table 6-2.

HCKT may be programmed as a general-purpose I/O pin (PC5) when the ESAI HCKT function is not being used.

6.2.12 High Frequency Clock for Receiver (HCKR)

HCKR is a bidirectional pin providing the receivers high frequency clock for the ESAI interface. The direction of this pin is determined by the RHCKD bit in the RCCR register. In the asynchronous mode (SYN=0), the HCKR pin operates as the high frequency clock input or output used by all the enabled receivers. In the synchronous mode (SYN=1), it operates as the serial flag 2 pin. For further information on pin mode and definition, see Table 6-9 and on receiver clock signals see Table 6-1.

When this pin is configured as serial flag pin, its direction is determined by the RHCKD bit in the RCCR register. When configured as the output flag OF2, this pin reflects the value of the OF2 bit in the SAICR register, and the data in the OF2 bit shows up at the pin synchronized to the frame sync being used by the transmitter and receiver sections. When configured as the input flag IF2, the data value at the pin is stored in the IF2 bit in the SAISR register, synchronized by the frame sync in normal mode or the slot in network mode.

HCKR may be programmed as a general-purpose I/O pin (PC2) when the ESAI HCKR function is not being used.

6.3 ESAI Programming Model

The ESAI can be viewed as five control registers, one status register, six transmit data registers, four receive data registers, two transmit slot mask registers, two receive slot mask registers and a

special-purpose time slot register. The following paragraphs give detailed descriptions and operations of each bit in the ESAI registers.

The ESAI pins can also function as GPIO pins (Port C), described in Section 6.5, "GPIO - Pins and Registers".

6.3.1 ESAI Transmitter Clock Control Register (TCCR)

The read/write Transmitter Clock Control Register (TCCR) controls the ESAI transmitter clock generator bit and frame sync rates, the bit clock and high frequency clock sources and the directions of the HCKT, FST and SCKT signals. (See Figure 6-2). In the synchronous mode (SYN=1), the bit clock defined for the transmitter determines the receiver bit clock as well. TCCR also controls the number of words per frame for the serial data.

	11	10	9	8	7	6	5	4	3	2	1	0
X:$FFFFB6	TDC2	TDC1	TDC0	TPSR	TPM7	TPM6	TPM5	TPM4	TPM3	TPM2	TPM1	TPM0
	23	22	21	20	19	18	17	16	15	14	13	12
	THCKD	TFSD	TCKD	THCKP	TFSP	TCKP	TFP3	TFP2	TFP1	TFP0	TDC4	TDC3

Figure 6-2 TCCR Register

Hardware and software reset clear all the bits of the TCCR register.

The TCCR control bits are described in the following paragraphs.

6.3.1.1 TCCR Transmit Prescale Modulus Select (TPM7–TPM0) - Bits 0–7

The TPM7–TPM0 bits specify the divide ratio of the prescale divider in the ESAI transmitter clock generator. A divide ratio from 1 to 256 (TPM[7:0]=$00 to $FF) may be selected. The bit clock output is available at the transmit serial bit clock (SCKT) pin of the DSP. The bit clock output is also available internally for use as the bit clock to shift the transmit and receive shift registers. The ESAI transmit clock generator functional diagram is shown in Figure 6-3.

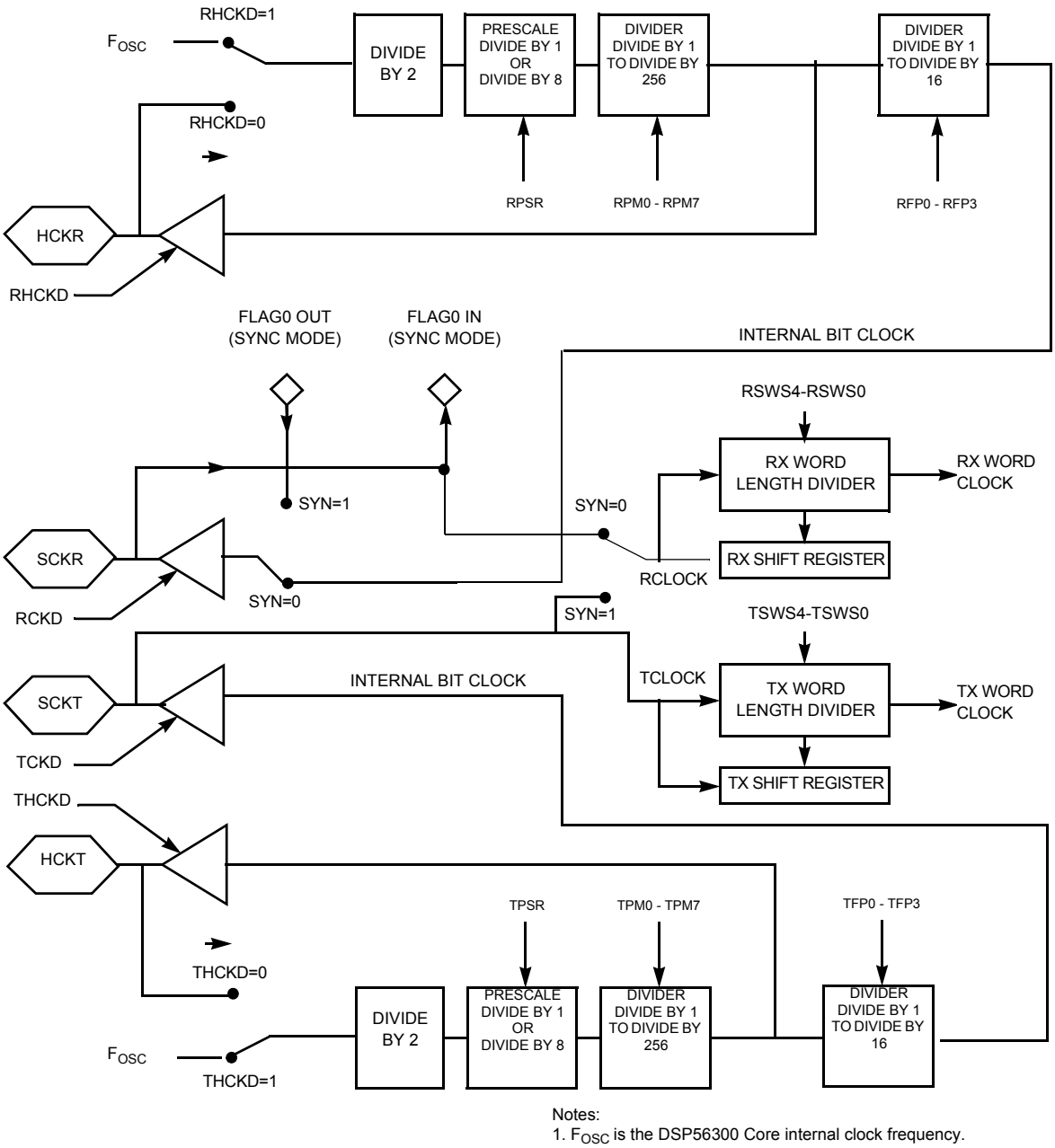


Figure 6-3 ESAI Clock Generator Functional Block Diagram

6.3.1.2 TCCR Transmit Prescaler Range (TPSR) - Bit 8

The TPSR bit controls a fixed divide-by-eight prescaler in series with the variable prescaler. This bit is used to extend the range of the prescaler for those cases where a slower bit clock is desired. When TPSR is set, the fixed prescaler is bypassed. When TPSR is cleared, the fixed divide-by-eight prescaler is

operational (see Figure 6-3). The maximum internally generated bit clock frequency is Fosc/4; the minimum internally generated bit clock frequency is Fosc/(2 x 8 x 256)=Fosc/4096.

NOTE

Do not use the combination TPSR=1 and TPM7-TPM0=$00, which causes synchronization problems when using the internal DSP clock as source (TCKD=1 or THCKD=1).

6.3.1.3 TCCR Tx Frame Rate Divider Control (TDC4–TDC0) - Bits 9–13

The TDC4–TDC0 bits control the divide ratio for the programmable frame rate dividers used to generate the transmitter frame clocks.

In network mode, this ratio may be interpreted as the number of words per frame minus one. The divide ratio may range from 2 to 32 (TDC[4:0]=00001 to 11111) for network mode. A divide ratio of one (TDC[4:0]=00000) in network mode is a special case (on-demand mode).

In normal mode, this ratio determines the word transfer rate. The divide ratio may range from 1 to 32 (TDC[4:0]=00000 to 11111) for normal mode. In normal mode, a divide ratio of 1 (TDC[4:0]=00000) provides continuous periodic data word transfers. A bit-length frame sync (TFSL=1) must be used in this case.

The ESAI frame sync generator functional diagram is shown in Figure 6-4.

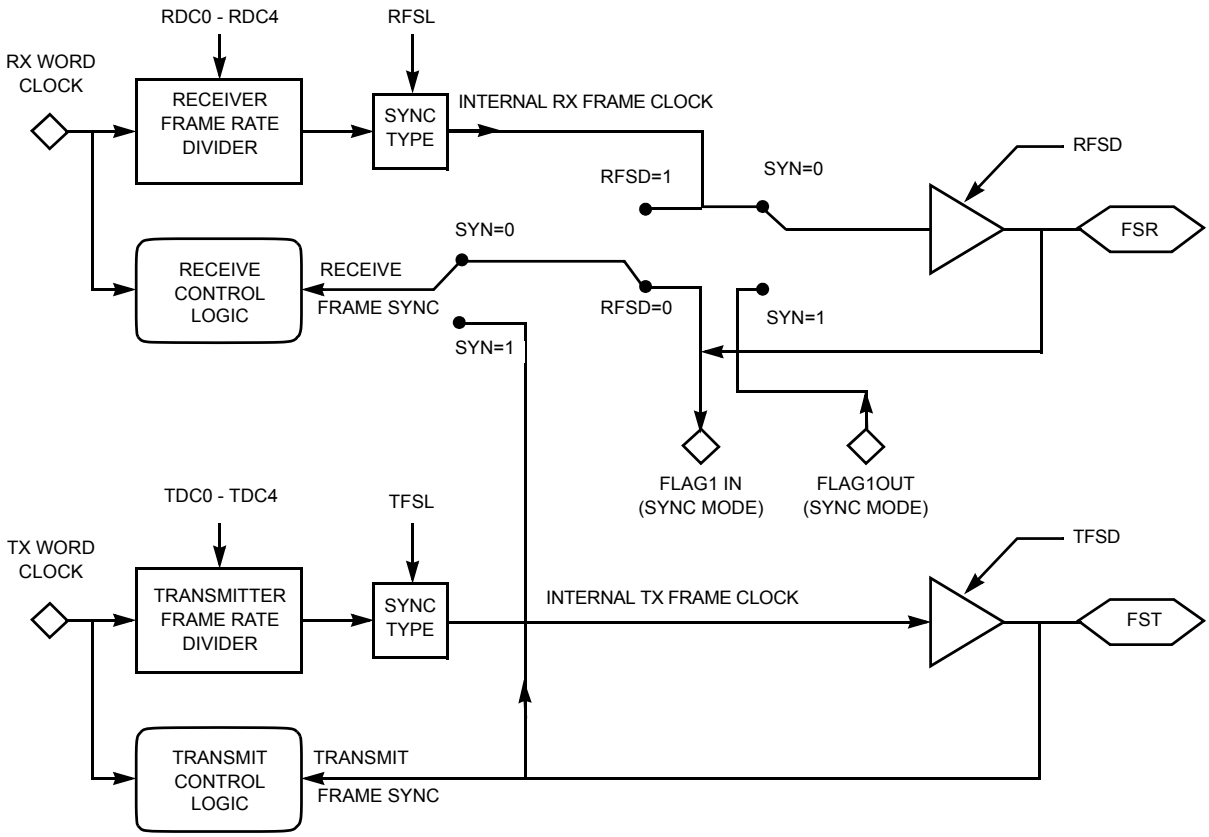


Figure 6-4 ESAI Frame Sync Generator Functional Block Diagram

6.3.1.4 TCCR Tx High Frequency Clock Divider (TFP3-TFP0) - Bits 14–17

The TFP3–TFP0 bits control the divide ratio of the transmitter high frequency clock to the transmitter serial bit clock when the source of the high frequency clock and the bit clock is the internal DSP clock. When the HCKT input is being driven from an external high frequency clock, the TFP3-TFP0 bits specify an additional division ratio in the clock divider chain. See Table 6-3 for the specification of the divide ratio. The ESAI high frequency clock generator functional diagram is shown in Figure 6-3.

Table 6-3 Transmitter High Frequency Clock Divider

TFP3-TFP0	Divide Ratio
$0	1
$1	2
$2	3
$3	4
...	...
$F	16

6.3.1.5 TCCR Transmit Clock Polarity (TCKP) - Bit 18

The Transmitter Clock Polarity (TCKP) bit controls on which bit clock edge data and frame sync are clocked out and latched in. If TCKP is cleared the data and the frame sync are clocked out on the rising edge of the transmit bit clock and latched in on the falling edge of the transmit bit clock. If TCKP is set the falling edge of the transmit clock is used to clock the data out and frame sync and the rising edge of the transmit clock is used to latch the data and frame sync in.

6.3.1.6 TCCR Transmit Frame Sync Polarity (TFSP) - Bit 19

The Transmitter Frame Sync Polarity (TFSP) bit determines the polarity of the transmit frame sync signal. When TFSP is cleared, the frame sync signal polarity is positive (i.e the frame start is indicated by a high level on the frame sync pin). When TFSP is set, the frame sync signal polarity is negative (i.e the frame start is indicated by a low level on the frame sync pin).

6.3.1.7 TCCR Transmit High Frequency Clock Polarity (THCKP) - Bit 20

The Transmitter High Frequency Clock Polarity (THCKP) bit controls on which bit clock edge data and frame sync are clocked out and latched in. If THCKP is cleared the data and the frame sync are clocked out on the rising edge of the transmit bit clock and latched in on the falling edge of the transmit bit clock. If THCKP is set the falling edge of the transmit clock is used to clock the data out and frame sync and the rising edge of the transmit clock is used to latch the data and frame sync in.

6.3.1.8 TCCR Transmit Clock Source Direction (TCKD) - Bit 21

The Transmitter Clock Source Direction (TCKD) bit selects the source of the clock signal used to clock the transmit shift registers in the asynchronous mode (SYN=0) and the transmit shift registers and the receive shift registers in the synchronous mode (SYN=1). When TCKD is set, the internal clock source becomes the bit clock for the transmit shift registers and word length divider and is the output on the SCKT pin. When TCKD is cleared, the clock source is external; the internal clock generator is disconnected from the SCKT pin, and an external clock source may drive this pin. See Table 6-2.

6.3.1.9 TCCR Transmit Frame Sync Signal Direction (TFSD) - Bit 22

TFSD controls the direction of the FST pin. When TFSD is cleared, FST is an input; when TFSD is set, FST is an output. See Table 6-2.

6.3.1.10 TCCR Transmit High Frequency Clock Direction (THCKD) - Bit 23

THCKD controls the direction of the HCKT pin. When THCKD is cleared, HCKT is an input; when THCKD is set, HCKT is an output. See Table 6-2.

6.3.2 ESAI Transmit Control Register (TCR)

The read/write Transmit Control Register (TCR) controls the ESAI transmitter section. Interrupt enable bits for the transmitter section are provided in this control register. Operating modes are also selected in this register. See Figure 6-5.

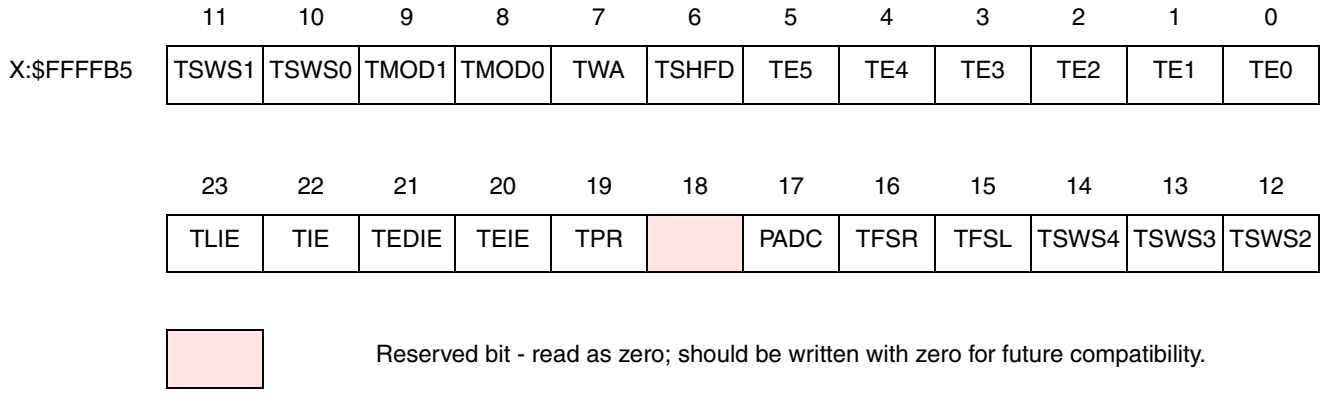


Figure 6-5 TCR Register

Hardware and software reset clear all the bits in the TCR register.

The TCR bits are described in the following paragraphs.

6.3.2.1 TCR ESAI Transmit 0 Enable (TE0) - Bit 0

TE0 enables the transfer of data from TX0 to the transmit shift register #0. When TE0 is set and a frame sync is detected, the transmit #0 portion of the ESAI is enabled for that frame. When TE0 is cleared, the transmitter #0 is disabled after completing transmission of data currently in the ESAI transmit shift register. The SDO0 output is tri-stated, and any data present in TX0 is not transmitted (i.e., data can be written to TX0 with TE0 cleared; but data is not transferred to the transmit shift register #0).

The normal mode transmit enable sequence is to write data to one or more transmit data registers before setting TEx. The normal transmit disable sequence is to clear TEx, TIE and TEIE after TDE equals one.

In the network mode, the operation of clearing TE0 and setting it again disables the transmitter #0 after completing transmission of the current data word until the beginning of the next frame. During that time period, the SDO0 pin remains in the high-impedance state.The on-demand mode transmit enable sequence can be the same as the normal mode, or TE0 can be left enabled.

6.3.2.2 TCR ESAI Transmit 1 Enable (TE1) - Bit 1

TE1 enables the transfer of data from TX1 to the transmit shift register #1. When TE1 is set and a frame sync is detected, the transmit #1 portion of the ESAI is enabled for that frame. When TE1 is cleared, the transmitter #1 is disabled after completing transmission of data currently in the ESAI transmit shift register. The SDO1 output is tri-stated, and any data present in TX1 is not transmitted (i.e., data can be written to TX1 with TE1 cleared; but data is not transferred to the transmit shift register #1).

The normal mode transmit enable sequence is to write data to one or more transmit data registers before setting TEx. The normal transmit disable sequence is to clear TEx, TIE and TEIE after TDE equals one.

In the network mode, the operation of clearing TE1 and setting it again disables the transmitter #1 after completing transmission of the current data word until the beginning of the next frame. During that time period, the SDO1 pin remains in the high-impedance state. The on-demand mode transmit enable sequence can be the same as the normal mode, or TE1 can be left enabled.

6.3.2.3 TCR ESAI Transmit 2 Enable (TE2) - Bit 2

TE2 enables the transfer of data from TX2 to the transmit shift register #2. When TE2 is set and a frame sync is detected, the transmit #2 portion of the ESAI is enabled for that frame. When TE2 is cleared, the transmitter #2 is disabled after completing transmission of data currently in the ESAI transmit shift register. Data can be written to TX2 when TE2 is cleared but the data is not transferred to the transmit shift register #2.

The SDO2/SDI3 pin is the data input pin for RX3 if TE2 is cleared and RE3 in the RCR register is set. If both RE3 and TE2 are cleared the transmitter and receiver are disabled, and the pin is tri-stated. Both RE3 and TE2 should not be set at the same time.

The normal mode transmit enable sequence is to write data to one or more transmit data registers before setting TEx. The normal transmit disable sequence is to clear TEx, TIE and TEIE after TDE equals one.

In the network mode, the operation of clearing TE2 and setting it again disables the transmitter #2 after completing transmission of the current data word until the beginning of the next frame. During that time period, the SDO2/SDI3 pin remains in the high-impedance state. The on-demand mode transmit enable sequence can be the same as the normal mode, or TE2 can be left enabled.

6.3.2.4 TCR ESAI Transmit 3 Enable (TE3) - Bit 3

TE3 enables the transfer of data from TX3 to the transmit shift register #3. When TE3 is set and a frame sync is detected, the transmit #3 portion of the ESAI is enabled for that frame. When TE3 is cleared, the transmitter #3 is disabled after completing transmission of data currently in the ESAI transmit shift register. Data can be written to TX3 when TE3 is cleared but the data is not transferred to the transmit shift register #3.

The SDO3/SDI2 pin is the data input pin for RX2 if TE3 is cleared and RE2 in the RCR register is set. If both RE2 and TE3 are cleared the transmitter and receiver are disabled, and the pin is tri-stated. Both RE2 and TE3 should not be set at the same time.

The normal mode transmit enable sequence is to write data to one or more transmit data registers before setting TEx. The normal transmit disable sequence is to clear TEx, TIE and TEIE after TDE equals one.

In the network mode, the operation of clearing TE3 and setting it again disables the transmitter #3 after completing transmission of the current data word until the beginning of the next frame. During that time period, the SDO3/SDI2 pin remains in the high-impedance state. The on-demand mode transmit enable sequence can be the same as the normal mode, or TE3 can be left enabled.

6.3.2.5 TCR ESAI Transmit 4 Enable (TE4) - Bit 4

TE4 enables the transfer of data from TX4 to the transmit shift register #4. When TE4 is set and a frame sync is detected, the transmit #4 portion of the ESAI is enabled for that frame. When TE4 is cleared, the

transmitter #4 is disabled after completing transmission of data currently in the ESAI transmit shift register. Data can be written to TX4 when TE4 is cleared but the data is not transferred to the transmit shift register #4.

The SDO4/SDI1 pin is the data input pin for RX1 if TE4 is cleared and RE1 in the RCR register is set. If both RE1 and TE4 are cleared the transmitter and receiver are disabled, and the pin is tri-stated. Both RE1 and TE4 should not be set at the same time.

The normal mode transmit enable sequence is to write data to one or more transmit data registers before setting TEx. The normal transmit disable sequence is to clear TEx, TIE and TEIE after TDE equals one.

In the network mode, the operation of clearing TE4 and setting it again disables the transmitter #4 after completing transmission of the current data word until the beginning of the next frame. During that time period, the SDO4/SDI1 pin remains in the high-impedance state. The on-demand mode transmit enable sequence can be the same as the normal mode, or TE4 can be left enabled.

6.3.2.6 TCR ESAI Transmit 5 Enable (TE5) - Bit 5

TE5 enables the transfer of data from TX5 to the transmit shift register #5. When TE5 is set and a frame sync is detected, the transmit #5 portion of the ESAI is enabled for that frame. When TE5 is cleared, the transmitter #5 is disabled after completing transmission of data currently in the ESAI transmit shift register. Data can be written to TX5 when TE5 is cleared but the data is not transferred to the transmit shift register #5.

The SDO5/SDI0 pin is the data input pin for RX0 if TE5 is cleared and RE0 in the RCR register is set. If both RE0 and TE5 are cleared the transmitter and receiver are disabled, and the pin is tri-stated. Both RE0 and TE5 should not be set at the same time.

The normal mode transmit enable sequence is to write data to one or more transmit data registers before setting TEx. The normal transmit disable sequence is to clear TEx, TIE and TEIE after TDE equals one.

In the network mode, the operation of clearing TE5 and setting it again disables the transmitter #5 after completing transmission of the current data word until the beginning of the next frame. During that time period, the SDO5/SDI0 pin remains in the high-impedance state. The on-demand mode transmit enable sequence can be the same as the normal mode, or TE5 can be left enabled.

6.3.2.7 TCR Transmit Shift Direction (TSHFD) - Bit 6

The TSHFD bit causes the transmit shift registers to shift data out MSB first when TSHFD equals zero or LSB first when TSHFD equals one (see Figure 6-13 and Figure 6-14).

6.3.2.8 TCR Transmit Word Alignment Control (TWA) - Bit 7

The Transmitter Word Alignment Control (TWA) bit defines the alignment of the data word in relation to the slot. This is relevant for the cases where the word length is shorter than the slot length. If TWA is cleared, the data word is left-aligned in the slot frame during transmission. If TWA is set, the data word is right-aligned in the slot frame during transmission.

Since the data word is shorter than the slot length, the data word is extended until achieving the slot length, according to the following rule:

1. If the data word is left-aligned (TWA=0), and zero padding is disabled (PADC=0), then the last data bit is repeated after the data word has been transmitted. If zero padding is enabled (PADC=1), zeroes are transmitted after the data word has been transmitted.
2. If the data word is right-aligned (TWA=1), and zero padding is disabled (PADC=0), then the first data bit is repeated before the transmission of the data word. If zero padding is enabled (PADC=1), zeroes are transmitted before the transmission of the data word.

6.3.2.9 TCR Transmit Network Mode Control (TMOD1-TMOD0) - Bits 8-9

The TMOD1 and TMOD0 bits are used to define the network mode of ESAI transmitters according to Table 6-4. In the normal mode, the frame rate divider determines the word transfer rate – one word is transferred per frame sync during the frame sync time slot, as shown in Figure 6-6. In network mode, it is possible to transfer a word for every time slot, as shown in Figure 6-6. For more details, see Section 6.4, "Operating Modes".

In order to comply with AC-97 specifications, TSWS4-TSWS0 should be set to 00011 (20-bit slot, 20-bit word length), TFSL and TFSR should be cleared, and TDC4-TDC0 should be set to $0C (13 words in frame). If TMOD[1:0]=$11 and the above recommendations are followed, the first slot and word will be 16 bits long, and the next 12 slots and words will be 20 bits long, as required by the AC97 protocol.

Table 6-4 Transmit Network Mode Selection

TMOD1	TMOD0	TDC4-TDC0	Transmitter Network Mode
0	0	$0-$1F	Normal Mode
0	1	$0	On-Demand Mode
0	1	$1-$1F	Network Mode
1	0	X	Reserved
1	1	$0C	AC97

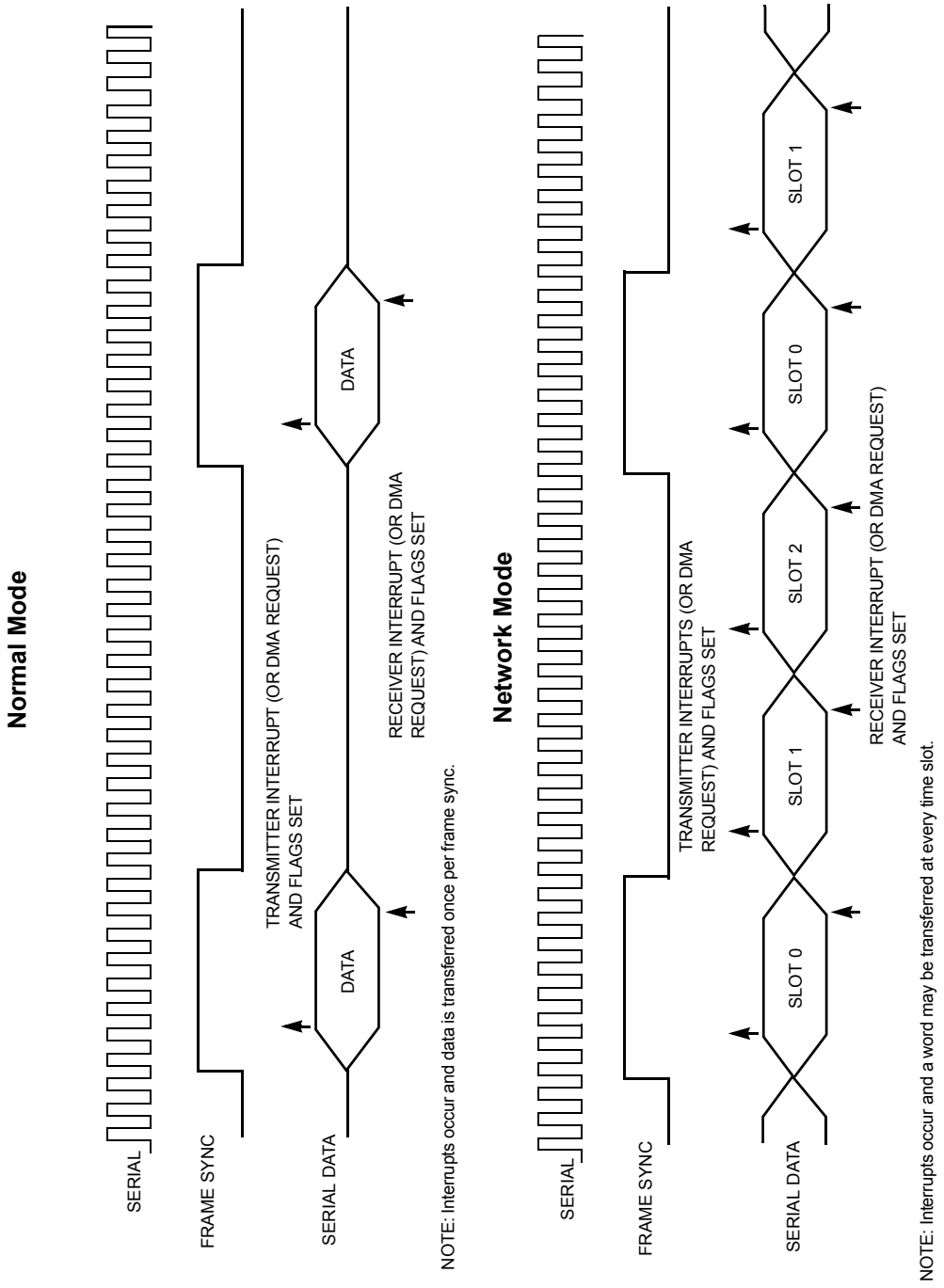


Figure 6-6 Normal and Network Operation

6.3.2.10 TCR Tx Slot and Word Length Select (TSWS4-TSWS0) - Bits 10-14

The TSWS4-TSWS0 bits are used to select the length of the slot and the length of the data words being transferred via the ESAI. The word length must be equal to or shorter than the slot length. The possible combinations are shown in Table 6-5. See also the ESAI data path programming model in Figure 6-13 and Figure 6-14.

Table 6-5 ESAI Transmit Slot and Word Length Selection

TSWS4	TSWS3	TSWS2	TSWS1	TSWS0	SLOT LENGTH	WORD LENGTH
0	0	0	0	0	8	8
0	0	1	0	0	12	8
0	0	0	0	1		12
0	1	0	0	0	16	8
0	0	1	0	1		12
0	0	0	1	0		16
0	1	1	0	0	20	8
0	1	0	0	1		12
0	0	1	1	0		16
0	0	0	1	1		20
1	0	0	0	0	24	8
0	1	1	0	1		12
0	1	0	1	0		16
0	0	1	1	1		20
1	1	1	1	0		24
1	1	0	0	0	32	8
1	0	1	0	1		12
1	0	0	1	0		16
0	1	1	1	1		20
1	1	1	1	1		24

Table 6-5 ESAI Transmit Slot and Word Length Selection (continued)

TSWS4	TSWS3	TSWS2	TSWS1	TSWS0	SLOT LENGTH	WORD LENGTH
0	1	0	1	1	Reserved	
0	1	1	1	0		
1	0	0	0	1		
1	0	0	1	1		
1	0	1	0	0		
1	0	1	1	0		
1	0	1	1	1		
1	1	0	0	1		
1	1	0	1	0		
1	1	0	1	1		
1	1	1	0	0		
1	1	1	0	1		

6.3.2.11 TCR Transmit Frame Sync Length (TFSL) - Bit 15

The TFSL bit selects the length of frame sync to be generated or recognized. If TFSL is cleared, a word-length frame sync is selected. If TFSL is set, a 1-bit clock period frame sync is selected. See Figure 6-7 for examples of frame length selection.

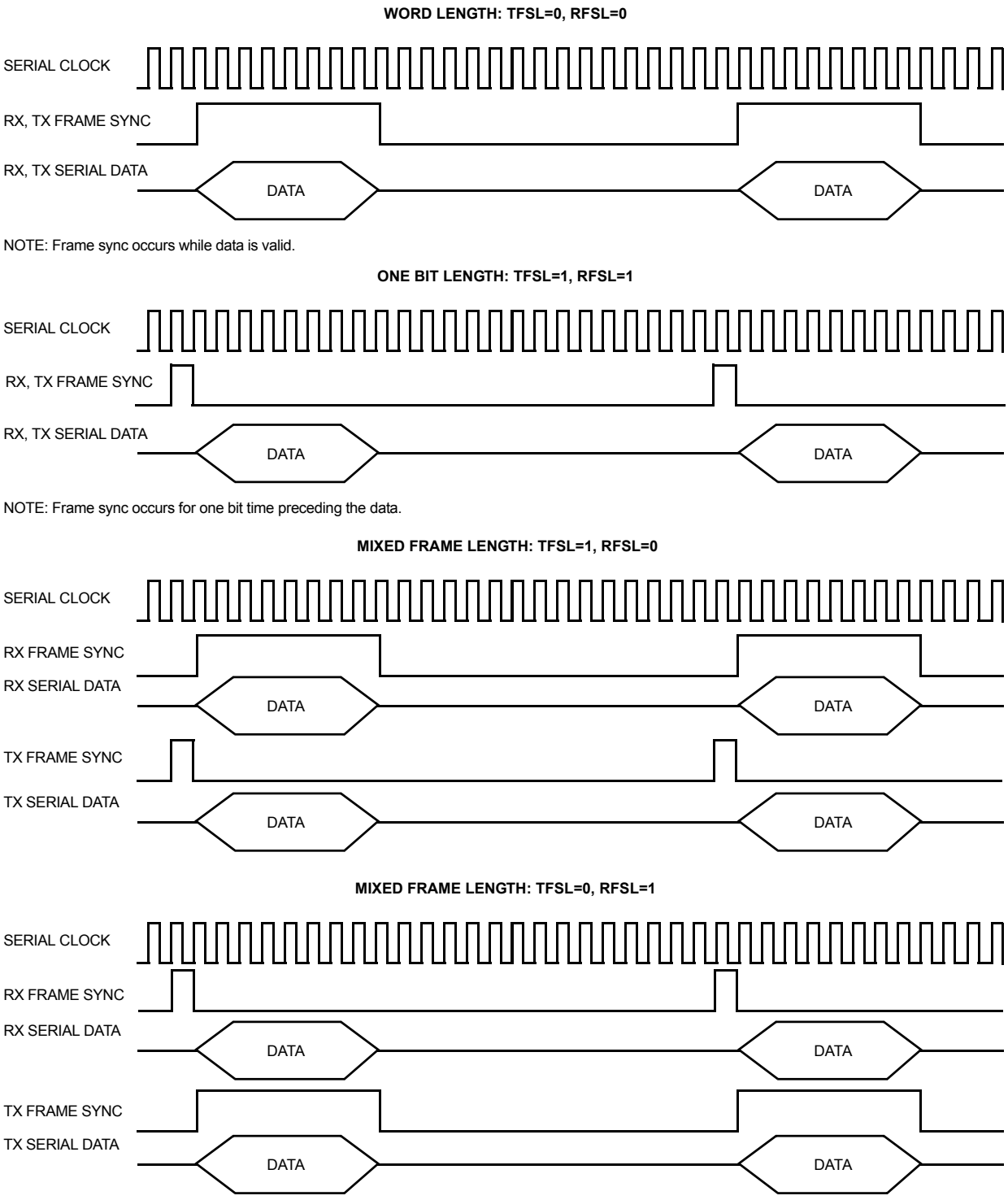


Figure 6-7 Frame Length Selection

6.3.2.12 TCR Transmit Frame Sync Relative Timing (TFSR) - Bit 16

TFSR determines the relative timing of the transmit frame sync signal as referred to the serial data lines, for a word length frame sync only (TFSL=0). When TFSR is cleared the word length frame sync occurs together with the first bit of the data word of the first slot. When TFSR is set the word length frame sync starts one serial clock cycle earlier (i.e together with the last bit of the previous data word).

6.3.2.13 TCR Transmit Zero Padding Control (PADC) - Bit 17

When PADC is cleared, zero padding is disabled. When PADC is set, zero padding is enabled. PADC, in conjunction with the TWA control bit, determines the way that padding is done for operating modes where the word length is less than the slot length. See the TWA bit description in Section 6.3.2.8, "TCR Transmit Word Alignment Control (TWA) - Bit 7" for more details.

Since the data word is shorter than the slot length, the data word is extended until achieving the slot length, according to the following rule:

1. If the data word is left-aligned (TWA=0), and zero padding is disabled (PADC=0), then the last data bit is repeated after the data word has been transmitted. If zero padding is enabled (PADC=1), zeroes are transmitted after the data word has been transmitted.
2. If the data word is right-aligned (TWA=1), and zero padding is disabled (PADC=0), then the first data bit is repeated before the transmission of the data word. If zero padding is enabled (PADC=1), zeroes are transmitted before the transmission of the data word.

6.3.2.14 TCR Reserved Bit - Bits 18

This bit is reserved. It reads as zero, and it should be written with zero for future compatibility.

6.3.2.15 TCR Transmit Section Personal Reset (TPR) - Bit 19

The TPR control bit is used to put the transmitter section of the ESAI in the personal reset state. The receiver section is not affected. When TPR is cleared, the transmitter section may operate normally. When TPR is set, the transmitter section enters the personal reset state immediately. When in the personal reset state, the status bits are reset to the same state as after hardware reset. The control bits are not affected by the personal reset state. The transmitter data pins are tri-stated while in the personal reset state; if a stable logic level is desired, the transmitter data pins should be defined as GPIO outputs, or external pull-up or pull-down resistors should be used. The transmitter clock outputs drive zeroes while in the personal reset state. Note that to leave the personal reset state by clearing TPR, the procedure described in Section 6.6, "ESAI Initialization Examples" should be followed.

6.3.2.16 TCR Transmit Exception Interrupt Enable (TEIE) - Bit 20

When TEIE is set, the DSP is interrupted when both TDE and TUE in the SAISR status register are set. When TEIE is cleared, this interrupt is disabled. Reading the SAISR status register followed by writing to all the data registers of the enabled transmitters clears TUE, thus clearing the pending interrupt.

6.3.2.17 TCR Transmit Even Slot Data Interrupt Enable (TEDIE) - Bit 21

The TEDIE control bit is used to enable the transmit even slot data interrupts. If TEDIE is set, the transmit even slot data interrupts are enabled. If TEDIE is cleared, the transmit even slot data interrupts are disabled. A transmit even slot data interrupt request is generated if TEDIE is set and the TEDE status flag in the SAISR status register is set. Even time slots are all even-numbered time slots (0, 2, 4, etc.) when operating in network mode. The zero time slot in the frame is marked by the frame sync signal and is considered to be even. Writing data to all the data registers of the enabled transmitters or to TSR clears the TEDE flag, thus servicing the interrupt.

Transmit interrupts with exception have higher priority than transmit even slot data interrupts, therefore if exception occurs (TUE is set) and TEIE is set, the ESAI requests an ESAI transmit data with exception interrupt from the interrupt controller.

6.3.2.18 TCR Transmit Interrupt Enable (TIE) - Bit 22

The DSP is interrupted when TIE and the TDE flag in the SAISR status register are set. When TIE is cleared, this interrupt is disabled. Writing data to all the data registers of the enabled transmitters or to TSR clears TDE, thus clearing the interrupt.

Transmit interrupts with exception have higher priority than normal transmit data interrupts, therefore if exception occurs (TUE is set) and TEIE is set, the ESAI requests an ESAI transmit data with exception interrupt from the interrupt controller.

6.3.2.19 TCR Transmit Last Slot Interrupt Enable (TLIE) - Bit 23

TLIE enables an interrupt at the beginning of last slot of a frame in network mode. When TLIE is set the DSP is interrupted at the start of the last slot in a frame in network mode regardless of the transmit mask register setting. When TLIE is cleared the transmit last slot interrupt is disabled. TLIE is disabled when TDC[4:0]=$00000 (on-demand mode). The use of the transmit last slot interrupt is described in Section 6.4.3, "ESAI Interrupt Requests".

6.3.3 ESAI Receive Clock Control Register (RCCR)

The read/write Receive Clock Control Register (RCCR) controls the ESAI receiver clock generator bit and frame sync rates, word length, and number of words per frame for the serial data. The RCCR control bits are described in the following paragraphs (see Figure 6-8).

	11	10	9	8	7	6	5	4	3	2	1	0
X:$FFFFB8	RDC2	RDC1	RDC0	RPSR	RPM7	RPM6	RPM5	RPM4	RPM3	RPM2	RPM1	RPM0
	23	22	21	20	19	18	17	16	15	14	13	12
	RHCKD	RFSD	RCKD	RHCKP	RFSP	RCKP	RFP3	RFP2	RFP1	RFP0	RDC4	RDC3

Figure 6-8 RCCR Register

Hardware and software reset clear all the bits of the RCCR register.

6.3.3.1 RCCR Receiver Prescale Modulus Select (RPM7–RPM0) - Bits 7–0

The RPM7–RPM0 bits specify the divide ratio of the prescale divider in the ESAI receiver clock generator. A divide ratio from 1 to 256 (RPM[7:0]=$00 to $FF) may be selected. The bit clock output is available at the receiver serial bit clock (SCKR) pin of the DSP. The bit clock output is also available internally for use as the bit clock to shift the receive shift registers. The ESAI receive clock generator functional diagram is shown in Figure 6-3.

6.3.3.2 RCCR Receiver Prescaler Range (RPSR) - Bit 8

The RPSR controls a fixed divide-by-eight prescaler in series with the variable prescaler. This bit is used to extend the range of the prescaler for those cases where a slower bit clock is desired. When RPSR is set, the fixed prescaler is bypassed. When RPSR is cleared, the fixed divide-by-eight prescaler is operational (see Figure 6-3). The maximum internally generated bit clock frequency is Fosc/4, the minimum internally generated bit clock frequency is Fosc/(2 x 8 x 256)=Fosc/4096.

NOTE

Do not use the combination RPSR=1 and RPM7-RPM0=$00, which causes synchronization problems when using the internal DSP clock as source (RHCKD=1 or RCKD=1).

6.3.3.3 RCCR Rx Frame Rate Divider Control (RDC4–RDC0) - Bits 9–13

The RDC4–RDC0 bits control the divide ratio for the programmable frame rate dividers used to generate the receiver frame clocks.

In network mode, this ratio may be interpreted as the number of words per frame minus one. The divide ratio may range from 2 to 32 (RDC[4:0]=00001 to 11111) for network mode. A divide ratio of one (RDC[4:0]=00000) in network mode is a special case (on-demand mode).

In normal mode, this ratio determines the word transfer rate. The divide ratio may range from 1 to 32 (RDC[4:0]=00000 to 11111) for normal mode. In normal mode, a divide ratio of one (RDC[4:0]=00000) provides continuous periodic data word transfers. A bit-length frame sync (RFSL=1) must be used in this case.

The ESAI frame sync generator functional diagram is shown in Figure 6-4.

6.3.3.4 RCCR Rx High Frequency Clock Divider (RFP3-RFP0) - Bits 14-17

The RFP3–RFP0 bits control the divide ratio of the receiver high frequency clock to the receiver serial bit clock when the source of the receiver high frequency clock and the bit clock is the internal DSP clock. When the HCKR input is being driven from an external high frequency clock, the RFP3-RFP0 bits specify an additional division ration in the clock divider chain. See Table 6-6 for the specification of the divide ratio. The ESAI high frequency generator functional diagram is shown in Figure 6-3.

Table 6-6 Receiver High Frequency Clock Divider

RFP3-RFP0	Divide Ratio
$0	1
$1	2
$2	3
$3	4
...	...
$F	16

6.3.3.5 RCCR Receiver Clock Polarity (RCKP) - Bit 18

The Receiver Clock Polarity (RCKP) bit controls on which bit clock edge data and frame sync are clocked out and latched in. If RCKP is cleared the data and the frame sync are clocked out on the rising edge of the receive bit clock and the frame sync is latched in on the falling edge of the receive bit clock. If RCKP is set the falling edge of the receive clock is used to clock the data and frame sync out and the rising edge of the receive clock is used to latch the frame sync in.

6.3.3.6 RCCR Receiver Frame Sync Polarity (RFSP) - Bit 19

The Receiver Frame Sync Polarity (RFSP) determines the polarity of the receive frame sync signal. When RFSP is cleared the frame sync signal polarity is positive (i.e the frame start is indicated by a high level on the frame sync pin). When RFSP is set the frame sync signal polarity is negative (i.e the frame start is indicated by a low level on the frame sync pin).

6.3.3.7 RCCR Receiver High Frequency Clock Polarity (RHCKP) - Bit 20

The Receiver High Frequency Clock Polarity (RHCKP) bit controls on which bit clock edge data and frame sync are clocked out and latched in. If RHCKP is cleared the data and the frame sync are clocked out on the rising edge of the receive bit clock and the frame sync is latched in on the falling edge of the receive bit clock. If RHCKP is set the falling edge of the receive clock is used to clock the data and frame sync out and the rising edge of the receive clock is used to latch the frame sync in.

6.3.3.8 RCCR Receiver Clock Source Direction (RCKD) - Bit 21

The Receiver Clock Source Direction (RCKD) bit selects the source of the clock signal used to clock the receive shift register in the asynchronous mode (SYN=0) and the IF0/OF0 flag direction in the synchronous mode (SYN=1).

In the asynchronous mode when RCKD is set, the internal clock source becomes the bit clock for the receive shift registers and word length divider, and is the output on the SCKR pin. In the asynchronous mode when RCKD is cleared, the clock source is external; the internal clock generator is disconnected from the SCKR pin, and an external clock source may drive this pin.

In the synchronous mode when RCKD is set, the SCKR pin becomes the OF0 output flag. If RCKD is cleared, then the SCKR pin becomes the IF0 input flag. See Table 6-1 and Table 6-7.

Table 6-7 SCKR Pin Definition Table

Control Bits		SCKR PIN
SYN	RCKD	
0	0	SCKR input
0	1	SCKR output
1	0	IF0
1	1	OF0

6.3.3.9 RCCR Receiver Frame Sync Signal Direction (RFSD) - Bit 22

The Receiver Frame Sync Signal Direction (RFSD) bit selects the source of the receiver frame sync signal when in the asynchronous mode (SYN=0), and the IF1/OF1/Transmitter Buffer Enable flag direction in the synchronous mode (SYN=1).

In the asynchronous mode when RFSD is set, the internal clock generator becomes the source of the receiver frame sync, and is the output on the FSR pin. In the asynchronous mode when RFSD is cleared, the receiver frame sync source is external; the internal clock generator is disconnected from the FSR pin, and an external clock source may drive this pin.

In the synchronous mode when RFSD is set, the FSR pin becomes the OF1 output flag or the Transmitter Buffer Enable, according to the TEBE control bit. If RFSD is cleared, then the FSR pin becomes the IF1 input flag. See Table 6-1 and Table 6-8.

Table 6-8 FSR Pin Definition Table

Control Bits			FSR Pin
SYN	TEBE	RFSD	
0	X	0	FSR input
0	X	1	FSR output
1	0	0	IF1
1	0	1	OF1
1	1	0	reserved
1	1	1	Transmitter Buffer Enable

6.3.3.10 RCCR Receiver High Frequency Clock Direction (RHCKD) - Bit 23

The Receiver High Frequency Clock Direction (RHCKD) bit selects the source of the receiver high frequency clock when in the asynchronous mode (SYN=0), and the IF2/OF2 flag direction in the synchronous mode (SYN=1).

In the asynchronous mode when RHCKD is set, the internal clock generator becomes the source of the receiver high frequency clock, and is the output on the HCKR pin. In the asynchronous mode when RHCKD is cleared, the receiver high frequency clock source is external; the internal clock generator is disconnected from the HCKR pin, and an external clock source may drive this pin.

When RHCKD is cleared, HCKR is an input; when RHCKD is set, HCKR is an output.

In the synchronous mode when RHCKD is set, the HCKR pin becomes the OF2 output flag. If RHCKD is cleared, then the HCKR pin becomes the IF2 input flag. See Table 6-1 and Table 6-9.

Table 6-9 HCKR Pin Definition Table

Control Bits		HCKR PIN
SYN	RHCKD	
0	0	HCKR input
0	1	HCKR output
1	0	IF2
1	1	OF2

6.3.4 ESAI Receive Control Register (RCR)

The read/write Receive Control Register (RCR) controls the ESAI receiver section. Interrupt enable bits for the receivers are provided in this control register. The receivers are enabled in this register (0,1,2 or 3

receivers can be enabled) if the input data pin is not used by a transmitter. Operating modes are also selected in this register.

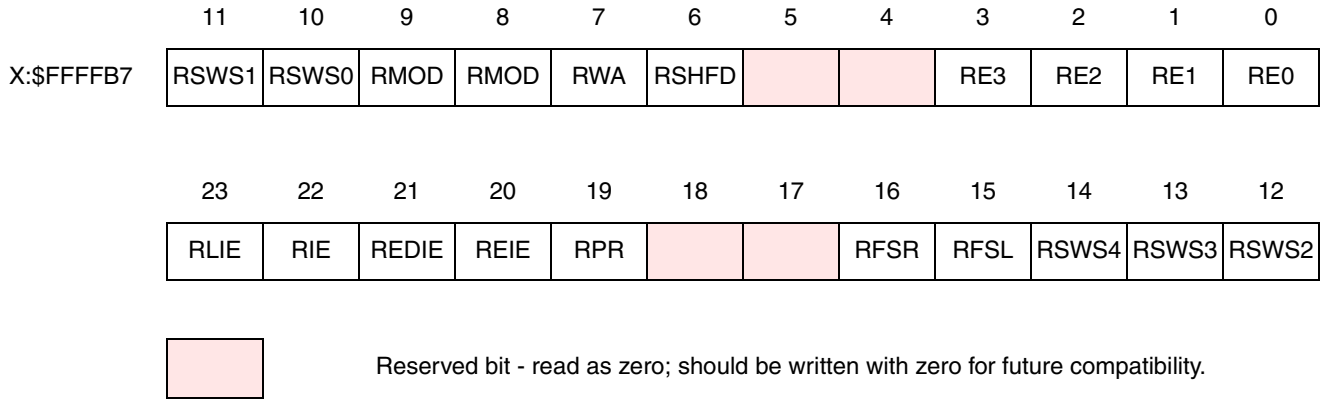


Figure 6-9 RCR Register

Hardware and software reset clear all the bits in the RCR register.

The ESAI RCR bits are described in the following paragraphs.

6.3.4.1 RCR ESAI Receiver 0 Enable (RE0) - Bit 0

When RE0 is set and TE5 is cleared, the ESAI receiver 0 is enabled and samples data at the SDO5/SDI0 pin. TX5 and RX0 should not be enabled at the same time (RE0=1 and TE5=1). When RE0 is cleared, receiver 0 is disabled by inhibiting data transfer into RX0. If this bit is cleared while receiving a data word, the remainder of the word is shifted in and transferred to the RX0 data register.

If RE0 is set while some of the other receivers are already in operation, the first data word received in RX0 will be invalid and must be discarded.

6.3.4.2 RCR ESAI Receiver 1 Enable (RE1) - Bit 1

When RE1 is set and TE4 is cleared, the ESAI receiver 1 is enabled and samples data at the SDO4/SDI1 pin. TX4 and RX1 should not be enabled at the same time (RE1=1 and TE4=1). When RE1 is cleared, receiver 1 is disabled by inhibiting data transfer into RX1. If this bit is cleared while receiving a data word, the remainder of the word is shifted in and transferred to the RX1 data register.

If RE1 is set while some of the other receivers are already in operation, the first data word received in RX1 will be invalid and must be discarded.

6.3.4.3 RCR ESAI Receiver 2 Enable (RE2) - Bit 2

When RE2 is set and TE3 is cleared, the ESAI receiver 2 is enabled and samples data at the SDO3/SDI2 pin. TX3 and RX2 should not be enabled at the same time (RE2=1 and TE3=1). When RE2 is cleared, receiver 2 is disabled by inhibiting data transfer into RX2. If this bit is cleared while receiving a data word, the remainder of the word is shifted in and transferred to the RX2 data register.

If RE2 is set while some of the other receivers are already in operation, the first data word received in RX2 will be invalid and must be discarded.

6.3.4.4 RCR ESAI Receiver 3 Enable (RE3) - Bit 3

When RE3 is set and TE2 is cleared, the ESAI receiver 3 is enabled and samples data at the SDO2/SDI3 pin. TX2 and RX3 should not be enabled at the same time (RE3=1 and TE2=1). When RE3 is cleared, receiver 3 is disabled by inhibiting data transfer into RX3. If this bit is cleared while receiving a data word, the remainder of the word is shifted in and transferred to the RX3 data register.

If RE3 is set while some of the other receivers are already in operation, the first data word received in RX3 will be invalid and must be discarded.

6.3.4.5 RCR Reserved Bits - Bits 4-5, 17-18

These bits are reserved. They read as zero, and they should be written with zero for future compatibility.

6.3.4.6 RCR Receiver Shift Direction (RSHFD) - Bit 6

The RSHFD bit causes the receiver shift registers to shift data in MSB first when RSHFD is cleared or LSB first when RSHFD is set (see Figure 6-13 and Figure 6-14).

6.3.4.7 RCR Receiver Word Alignment Control (RWA) - Bit 7

The Receiver Word Alignment Control (RWA) bit defines the alignment of the data word in relation to the slot. This is relevant for the cases where the word length is shorter than the slot length. If RWA is cleared, the data word is assumed to be left-aligned in the slot frame. If RWA is set, the data word is assumed to be right-aligned in the slot frame.

If the data word is shorter than the slot length, the data bits which are not in the data word field are ignored.

For data word lengths of less than 24 bits, the data word is right-extended with zeroes before being stored in the receive data registers.

6.3.4.8 RCR Receiver Network Mode Control (RMOD1-RMOD0) - Bits 8-9

The RMOD1 and RMOD0 bits are used to define the network mode of the ESAI receivers according to Table 6-10. In the normal mode, the frame rate divider determines the word transfer rate – one word is transferred per frame sync during the frame sync time slot, as shown in Figure 6-6. In network mode, it is possible to transfer a word for every time slot, as shown in Figure 6-6. For more details, see Section 6.4, "Operating Modes".

In order to comply with AC-97 specifications, RSWS4-RSWS0 should be set to 00011 (20-bit slot, 20-bit word), RFSL and RFSR should be cleared, and RDC4-RDC0 should be set to $0C (13 words in frame).

Table 6-10 ESAI Receive Network Mode Selection

RMOD1	RMOD0	RDC4-RDC0	Receiver Network Mode
0	0	$0-$1F	Normal Mode
0	1	$0	On-Demand Mode
0	1	$1-$1F	Network Mode
1	0	X	Reserved
1	1	$0C	AC97

6.3.4.9 RCR Receiver Slot and Word Select (RSWS4-RSWS0) - Bits 10-14

The RSWS4-RSWS0 bits are used to select the length of the slot and the length of the data words being received via the ESAI. The word length must be equal to or shorter than the slot length. The possible combinations are shown in Table 6-11. See also the ESAI data path programming model in Figure 6-13 and Figure 6-14.

Table 6-11 ESAI Receive Slot and Word Length Selection

RSWS4	RSWS3	RSWS2	RSWS1	RSWS0	SLOT LENGTH	WORD LENGTH
0	0	0	0	0	8	8
0	0	1	0	0	12	8
0	0	0	0	1		12
0	1	0	0	0	16	8
0	0	1	0	1		12
0	0	0	1	0		16
0	1	1	0	0	20	8
0	1	0	0	1		12
0	0	1	1	0		16
0	0	0	1	1		20
1	0	0	0	0	24	8
0	1	1	0	1		12
0	1	0	1	0		16
0	0	1	1	1		20
1	1	1	1	0		24

Table 6-11 ESAI Receive Slot and Word Length Selection (continued)

RSWS4	RSWS3	RSWS2	RSWS1	RSWS0	SLOT LENGTH	WORD LENGTH
1	1	0	0	0	32	8
1	0	1	0	1		12
1	0	0	1	0		16
0	1	1	1	1		20
1	1	1	1	1		24
0	1	0	1	1	Reserved	
0	1	1	1	0		
1	0	0	0	1		
1	0	0	1	1		
1	0	1	0	0		
1	0	1	1	0		
1	0	1	1	1		
1	1	0	0	1		
1	1	0	1	0		
1	1	0	1	1		
1	1	1	0	0		
1	1	1	0	1		

6.3.4.10 RCR Receiver Frame Sync Length (RFSL) - Bit 15

The RFSL bit selects the length of the receive frame sync to be generated or recognized. If RFSL is cleared, a word-length frame sync is selected. If RFSL is set, a 1-bit clock period frame sync is selected. See Figure 6-7 for examples of frame length selection.

6.3.4.11 RCR Receiver Frame Sync Relative Timing (RFSR) - Bit 16

RFSR determines the relative timing of the receive frame sync signal as referred to the serial data lines, for a word length frame sync only. When RFSR is cleared the word length frame sync occurs together with the first bit of the data word of the first slot. When RFSR is set the word length frame sync starts one serial clock cycle earlier (i.e. together with the last bit of the previous data word).

6.3.4.12 RCR Receiver Section Personal Reset (RPR) - Bit 19

The RPR control bit is used to put the receiver section of the ESAI in the personal reset state. The transmitter section is not affected. When RPR is cleared, the receiver section may operate normally. When RPR is set, the receiver section enters the personal reset state immediately. When in the personal reset state, the status bits are reset to the same state as after hardware reset.The control bits are not affected by the personal reset state.The receiver data pins are disconnected while in the personal reset state. Note that to leave the personal reset state by clearing RPR, the procedure described in Section 6.6, "ESAI Initialization Examples" should be followed.

6.3.4.13 RCR Receive Exception Interrupt Enable (REIE) - Bit 20

When REIE is set, the DSP is interrupted when both RDF and ROE in the SAISR status register are set. When REIE is cleared, this interrupt is disabled. Reading the SAISR status register followed by reading the enabled receivers data registers clears ROE, thus clearing the pending interrupt.

6.3.4.14 RCR Receive Even Slot Data Interrupt Enable (REDIE) - Bit 21

The REDIE control bit is used to enable the receive even slot data interrupts. If REDIE is set, the receive even slot data interrupts are enabled. If REDIE is cleared, the receive even slot data interrupts are disabled. A receive even slot data interrupt request is generated if REDIE is set and the REDF status flag in the SAISR status register is set. Even time slots are all even-numbered time slots (0, 2, 4, etc.) when operating in network mode. The zero time slot is marked by the frame sync signal and is considered to be even. Reading all the data registers of the enabled receivers clears the REDF flag, thus servicing the interrupt.

Receive interrupts with exception have higher priority than receive even slot data interrupts, therefore if exception occurs (ROE is set) and REIE is set, the ESAI requests an ESAI receive data with exception interrupt from the interrupt controller.

6.3.4.15 RCR Receive Interrupt Enable (RIE) - Bit 22

The DSP is interrupted when RIE and the RDF flag in the SAISR status register are set. When RIE is cleared, this interrupt is disabled. Reading the receive data registers of the enabled receivers clears RDF, thus clearing the interrupt.

Receive interrupts with exception have higher priority than normal receive data interrupts, therefore if exception occurs (ROE is set) and REIE is set, the ESAI requests an ESAI receive data with exception interrupt from the interrupt controller.

6.3.4.16 RCR Receive Last Slot Interrupt Enable (RLIE) - Bit 23

RLIE enables an interrupt after the last slot of a frame ended in network mode only. When RLIE is set the DSP is interrupted after the last slot in a frame ended regardless of the receive mask register setting. When RLIE is cleared the receive last slot interrupt is disabled. Hardware and software reset clear RLIE. RLIE is disabled when RDC[4:0]=00000 (on-demand mode). The use of the receive last slot interrupt is described in Section 6.4.3, "ESAI Interrupt Requests".

6.3.5 ESAI Common Control Register (SAICR)

The read/write Common Control Register (SAICR) contains control bits for functions that affect both the receive and transmit sections of the ESAI.See Figure 6-10.

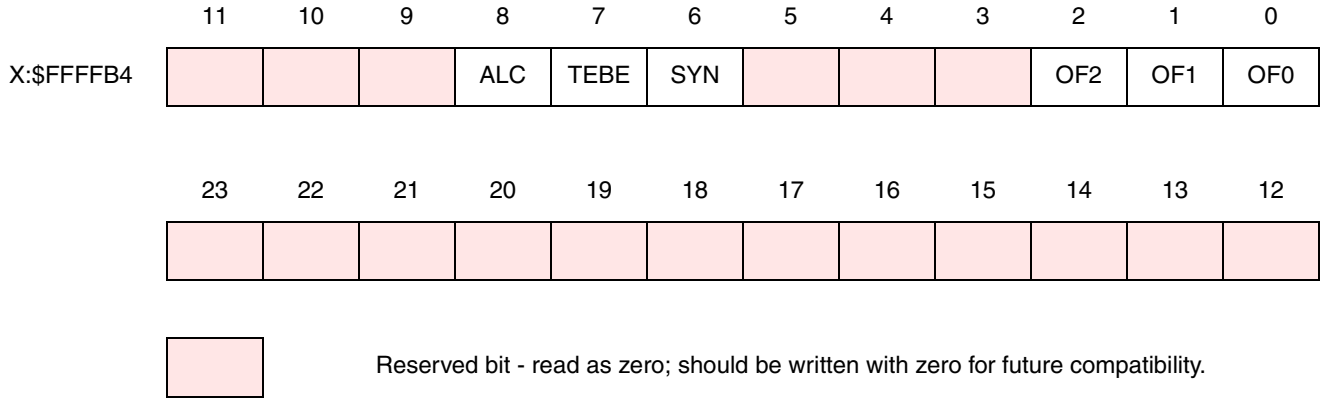


Figure 6-10 SAICR Register

Hardware and software reset clear all the bits in the SAICR register.

6.3.5.1 SAICR Serial Output Flag 0 (OF0) - Bit 0

The Serial Output Flag 0 (OF0) is a data bit used to hold data to be send to the OF0 pin. When the ESAI is in the synchronous clock mode (SYN=1), the SCKR pin is configured as the ESAI flag 0. If the receiver serial clock direction bit (RCKD) is set, the SCKR pin is the output flag OF0, and data present in the OF0 bit is written to the OF0 pin at the beginning of the frame in normal mode or at the beginning of the next time slot in network mode.

6.3.5.2 SAICR Serial Output Flag 1 (OF1) - Bit 1

The Serial Output Flag 1 (OF1) is a data bit used to hold data to be send to the OF1 pin. When the ESAI is in the synchronous clock mode (SYN=1), the FSR pin is configured as the ESAI flag 1. If the receiver frame sync direction bit (RFSD) is set and the TEBE bit is cleared, the FSR pin is the output flag OF1, and data present in the OF1 bit is written to the OF1 pin at the beginning of the frame in normal mode or at the beginning of the next time slot in network mode.

6.3.5.3 SAICR Serial Output Flag 2 (OF2) - Bit 2

The Serial Output Flag 2 (OF2) is a data bit used to hold data to be send to the OF2 pin. When the ESAI is in the synchronous clock mode (SYN=1), the HCKR pin is configured as the ESAI flag 2. If the receiver high frequency clock direction bit (RHCKD) is set, the HCKR pin is the output flag OF2, and data present in the OF2 bit is written to the OF2 pin at the beginning of the frame in normal mode or at the beginning of the next time slot in network mode.

6.3.5.4 SAICR Reserved Bits - Bits 3-5, 9-23

These bits are reserved. They read as zero, and they should be written with zero for future compatibility.

6.3.5.5 SAICR Synchronous Mode Selection (SYN) - Bit 6

The Synchronous Mode Selection (SYN) bit controls whether the receiver and transmitter sections of the ESAI operate synchronously or asynchronously with respect to each other (see Figure 6-11). When SYN is cleared, the asynchronous mode is chosen and independent clock and frame sync signals are used for the transmit and receive sections. When SYN is set, the synchronous mode is chosen and the transmit and receive sections use common clock and frame sync signals.

When in the synchronous mode (SYN=1), the transmit and receive sections use the transmitter section clock generator as the source of the clock and frame sync for both sections. Also, the receiver clock pins SCKR, FSR and HCKR now operate as I/O flags. See Table 6-7, Table 6-8 and Table 6-9 for the effects of SYN on the receiver clock pins.

6.3.5.6 SAICR Transmit External Buffer Enable (TEBE) - Bit 7

The Transmitter External Buffer Enable (TEBE) bit controls the function of the FSR pin when in the synchronous mode. If the ESAI is configured for operation in the synchronous mode (SYN=1), and TEBE is set while FSR pin is configured as an output (RFSD=1), the FSR pin functions as the transmitter external buffer enable control, to enable the use of an external buffers on the transmitter outputs. If TEBE is cleared then the FSR pin functions as the serial I/O flag 1. See Table 6-8 for a summary of the effects of TEBE on the FSR pin.

6.3.5.7 SAICR Alignment Control (ALC) - Bit 8

The ESAI is designed for 24-bit fractional data, thus shorter data words are left aligned to the MSB (bit 23). Some applications use 16-bit fractional data. In those cases, shorter data words may be left aligned to bit 15. The Alignment Control (ALC) bit supports these applications.

If ALC is set, transmitted and received words are left aligned to bit 15 in the transmit and receive shift registers. If ALC is cleared, transmitted and received word are left aligned to bit 23 in the transmit and receive shift registers.

NOTE

While ALC is set, 20-bit and 24-bit words may not be used, and word length control should specify 8-, 12- or 16-bit words, otherwise results are unpredictable.

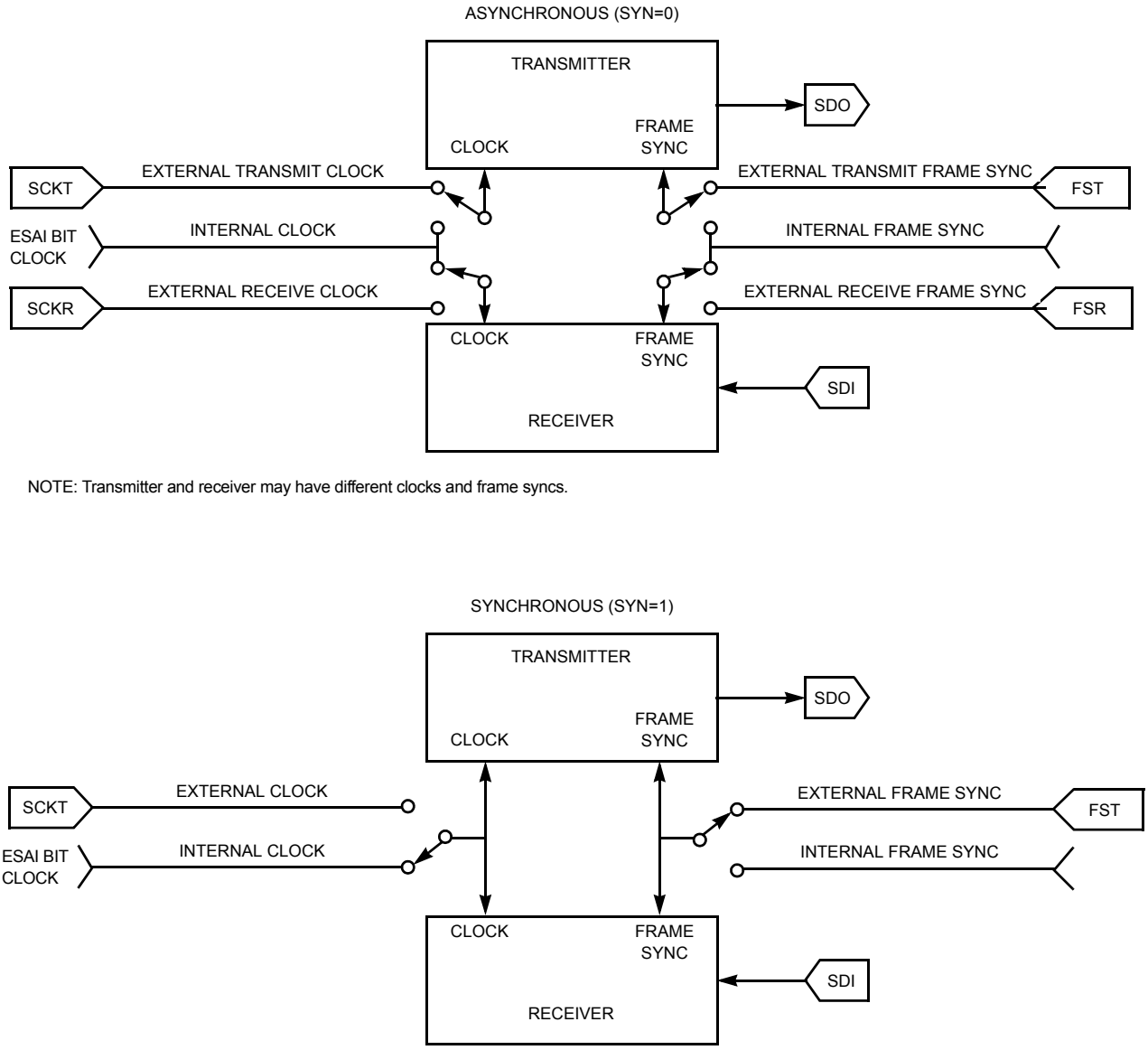


Figure 6-11 SAICR SYN Bit Operation

6.3.6 ESAI Status Register (SAISR)

The Status Register (SAISR) is a read-only status register used by the DSP to read the status and serial input flags of the ESAI. See Figure 6-12. The status bits are described in the following paragraphs.

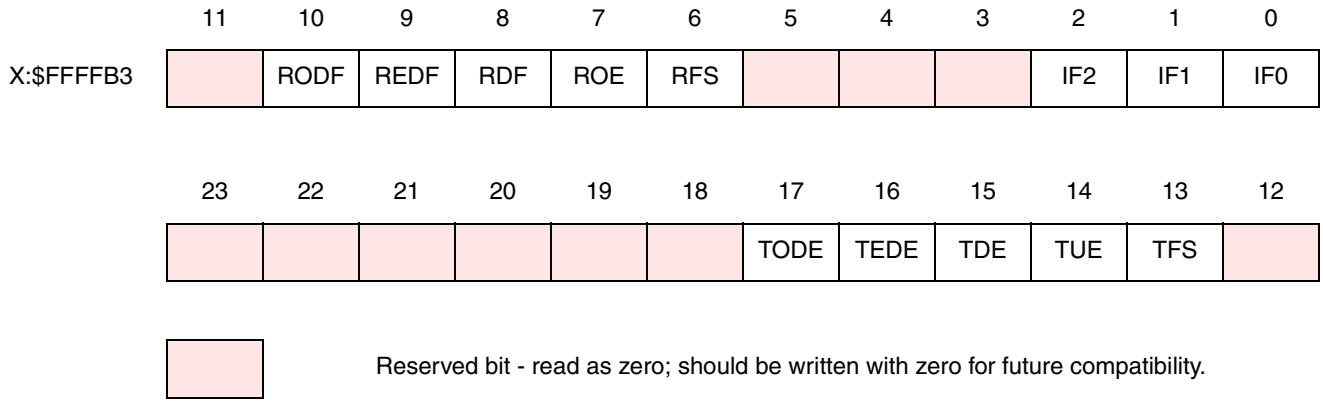


Figure 6-12 SAISR Register

6.3.6.1 SAISR Serial Input Flag 0 (IF0) - Bit 0

The IF0 bit is enabled only when the SCKR pin is defined as ESAI in the Port Control Register, SYN=1 and RCKD=0, indicating that SCKR is an input flag and the synchronous mode is selected. Data present on the SCKR pin is latched during reception of the first received data bit after frame sync is detected. The IF0 bit is updated with this data when the receiver shift registers are transferred into the receiver data registers. IF0 reads as a zero when it is not enabled. Hardware, software, ESAI individual, and STOP reset clear IF0.

6.3.6.2 SAISR Serial Input Flag 1 (IF1) - Bit 1

The IF1 bit is enabled only when the FSR pin is defined as ESAI in the Port Control Register, SYN =1, RFSD=0 and TEBE=0, indicating that FSR is an input flag and the synchronous mode is selected. Data present on the FSR pin is latched during reception of the first received data bit after frame sync is detected. The IF1 bit is updated with this data when the receiver shift registers are transferred into the receiver data registers. IF1 reads as a zero when it is not enabled. Hardware, software, ESAI individual, and STOP reset clear IF1.

6.3.6.3 SAISR Serial Input Flag 2 (IF2) - Bit 2

The IF2 bit is enabled only when the HCKR pin is defined as ESAI in the Port Control Register, SYN=1 and RHCKD=0, indicating that HCKR is an input flag and the synchronous mode is selected. Data present on the HCKR pin is latched during reception of the first received data bit after frame sync is detected. The IF2 bit is updated with this data when the receive shift registers are transferred into the receiver data registers. IF2 reads as a zero when it is not enabled. Hardware, software, ESAI individual, and STOP reset clear IF2.

6.3.6.4 SAISR Reserved Bits - Bits 3-5, 11-12, 18-23

These bits are reserved for future use. They read as zero.

6.3.6.5 SAISR Receive Frame Sync Flag (RFS) - Bit 6

When set, RFS indicates that a receive frame sync occurred during reception of the words in the receiver data registers. This indicates that the data words are from the first slot in the frame. When RFS is clear and a word is received, it indicates (only in the network mode) that the frame sync did not occur during reception of that word. RFS is cleared by hardware, software, ESAI individual, or STOP reset. RFS is valid only if at least one of the receivers is enabled (REx=1).

NOTE

In normal mode, RFS always reads as a one when reading data because there is only one time slot per frame – the “frame sync” time slot.

6.3.6.6 SAISR Receiver Overrun Error Flag (ROE) - Bit 7

The ROE flag is set when the serial receive shift register of an enabled receiver is full and ready to transfer to its receiver data register (RXx) and the register is already full (RDF=1). If REIE is set, an ESAI receive data with exception (overrun error) interrupt request is issued when ROE is set. Hardware, software, ESAI individual, and STOP reset clear ROE. ROE is also cleared by reading the SAISR with ROE set, followed by reading all the enabled receive data registers.

6.3.6.7 SAISR Receive Data Register Full (RDF) - Bit 8

RDF is set when the contents of the receive shift register of an enabled receiver is transferred to the respective receive data register. RDF is cleared when the DSP reads the receive data register of all enabled receivers or cleared by hardware, software, ESAI individual, or STOP reset. If RIE is set, an ESAI receive data interrupt request is issued when RDF is set.

6.3.6.8 SAISR Receive Even-Data Register Full (REDF) - Bit 9

When set, REDF indicates that the received data in the receive data registers of the enabled receivers have arrived during an even time slot when operating in the network mode. Even time slots are all even-numbered slots (0, 2, 4, 6, etc.). Time slots are numbered from zero to N-1, where N is the number of time slots in the frame. The zero time slot is considered even. REDF is set when the contents of the receive shift registers are transferred to the receive data registers. REDF is cleared when the DSP reads all the enabled receive data registers or cleared by hardware, software, ESAI individual, or STOP resets. If REDIE is set, an ESAI receive even slot data interrupt request is issued when REDF is set.

6.3.6.9 SAISR Receive Odd-Data Register Full (RODF) - Bit 10

When set, RODF indicates that the received data in the receive data registers of the enabled receivers have arrived during an odd time slot when operating in the network mode. Odd time slots are all odd-numbered slots (1, 3, 5, etc.). Time slots are numbered from zero to N-1, where N is the number of time slots in the frame. RODF is set when the contents of the receive shift registers are transferred to the receive data registers. RODF is cleared when the DSP reads all the enabled receive data registers or cleared by hardware, software, ESAI individual, or STOP resets.

6.3.6.10 SAISR Transmit Frame Sync Flag (TFS) - Bit 13

When set, TFS indicates that a transmit frame sync occurred in the current time slot. TFS is set at the start of the first time slot in the frame and cleared during all other time slots. Data written to a transmit data register during the time slot when TFS is set is transmitted (in network mode), if the transmitter is enabled, during the second time slot in the frame. TFS is useful in network mode to identify the start of a frame. TFS is cleared by hardware, software, ESAI individual, or STOP reset. TFS is valid only if at least one transmitter is enabled (i.e. one or more of TE0, TE1, TE2, TE3, TE4 and TE5 are set).

NOTE

In normal mode, TFS always reads as a one when transmitting data because there is only one time slot per frame – the “frame sync” time slot.

6.3.6.11 SAISR Transmit Underrun Error Flag (TUE) - Bit 14

TUE is set when at least one of the enabled serial transmit shift registers is empty (no new data to be transmitted) and a transmit time slot occurs. When a transmit underrun error occurs, the previous data (which is still present in the TX registers that were not written) is retransmitted. If TEIE is set, an ESAI transmit data with exception (underrun error) interrupt request is issued when TUE is set. Hardware, software, ESAI individual, and STOP reset clear TUE. TUE is also cleared by reading the SAISR with TUE set, followed by writing to all the enabled transmit data registers or to TSR.

6.3.6.12 SAISR Transmit Data Register Empty (TDE) - Bit 15

TDE is set when the contents of the transmit data register of all the enabled transmitters are transferred to the transmit shift registers; it is also set for a TSR disabled time slot period in network mode (as if data were being transmitted after the TSR was written). When set, TDE indicates that data should be written to all the TX registers of the enabled transmitters or to the time slot register (TSR). TDE is cleared when the DSP writes to all the transmit data registers of the enabled transmitters, or when the DSP writes to the TSR to disable transmission of the next time slot. If TIE is set, an ESAI transmit data interrupt request is issued when TDE is set. Hardware, software, ESAI individual, and STOP reset clear TDE.

6.3.6.13 SAISR Transmit Even-Data Register Empty (TEDE) - Bit 16

When set, TEDE indicates that the enabled transmitter data registers became empty at the beginning of an even time slot. Even time slots are all even-numbered slots (0, 2, 4, 6, etc.). Time slots are numbered from zero to N-1, where N is the number of time slots in the frame. The zero time slot is considered even. This flag is set when the contents of the transmit data register of all the enabled transmitters are transferred to the transmit shift registers; it is also set for a TSR disabled time slot period in network mode (as if data were being transmitted after the TSR was written). When set, TEDE indicates that data should be written to all the TX registers of the enabled transmitters or to the time slot register (TSR). TEDE is cleared when the DSP writes to all the transmit data registers of the enabled transmitters, or when the DSP writes to the TSR to disable transmission of the next time slot. If TIE is set, an ESAI transmit data interrupt request is issued when TEDE is set. Hardware, software, ESAI individual, and STOP reset clear TEDE.

6.3.6.14 SAISR Transmit Odd-Data Register Empty (TODE) - Bit 17

When set, TODE indicates that the enabled transmitter data registers became empty at the beginning of an odd time slot. Odd time slots are all odd-numbered slots (1, 3, 5, etc.). Time slots are numbered from zero to N-1, where N is the number of time slots in the frame. This flag is set when the contents of the transmit data register of all the enabled transmitters are transferred to the transmit shift registers; it is also set for a TSR disabled time slot period in network mode (as if data were being transmitted after the TSR was written). When set, TODE indicates that data should be written to all the TX registers of the enabled transmitters or to the time slot register (TSR). TODE is cleared when the DSP writes to all the transmit data registers of the enabled transmitters, or when the DSP writes to the TSR to disable transmission of the next time slot. If TIE is set, an ESAI transmit data interrupt request is issued when TODE is set. Hardware, software, ESAI individual, and STOP reset clear TODE.

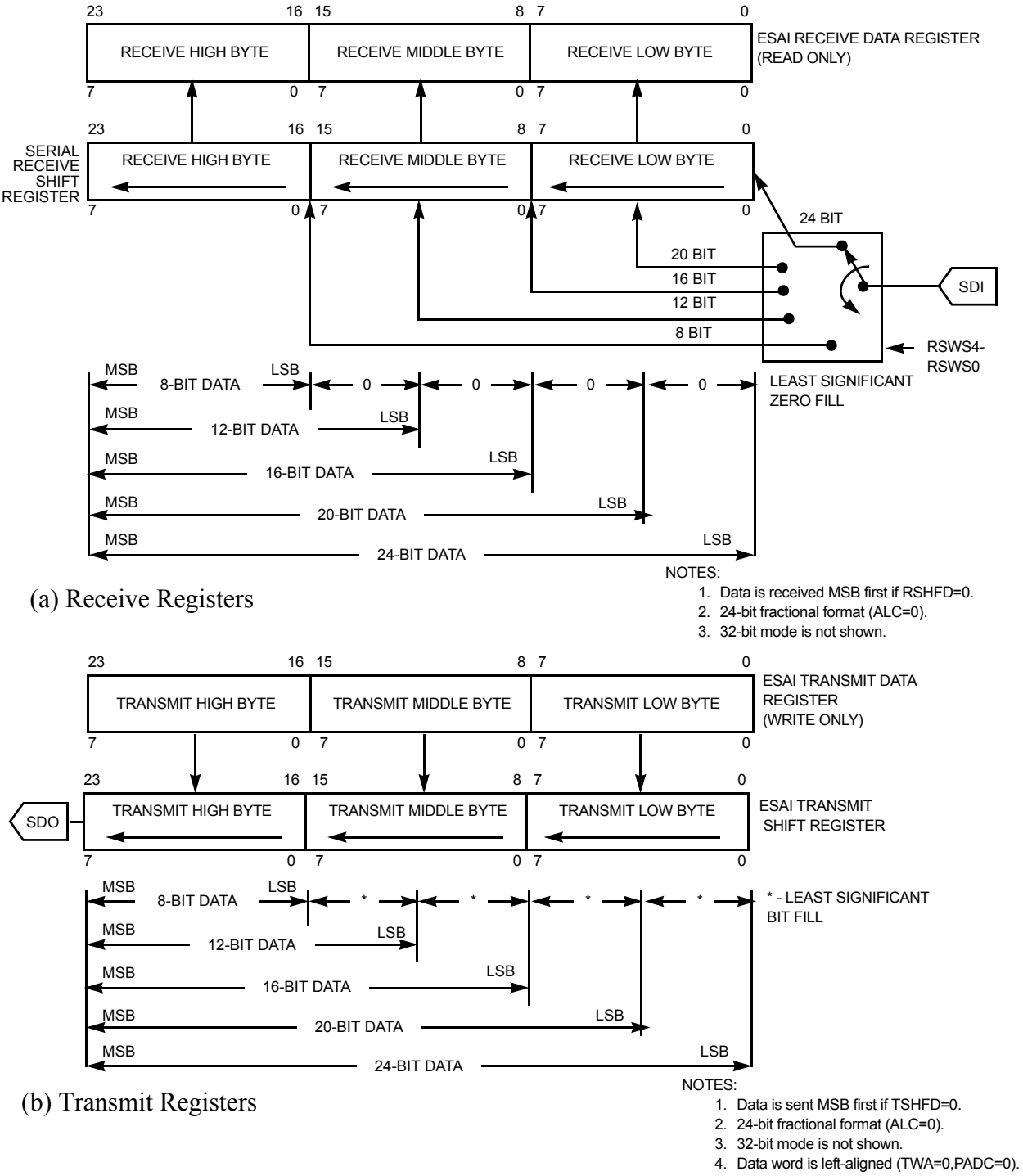
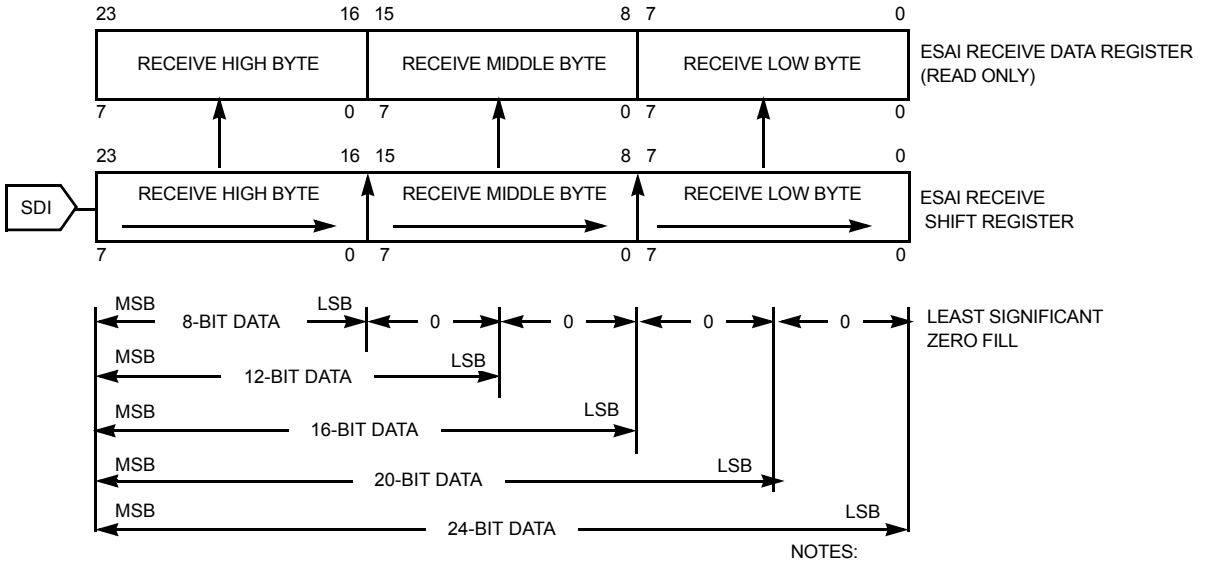
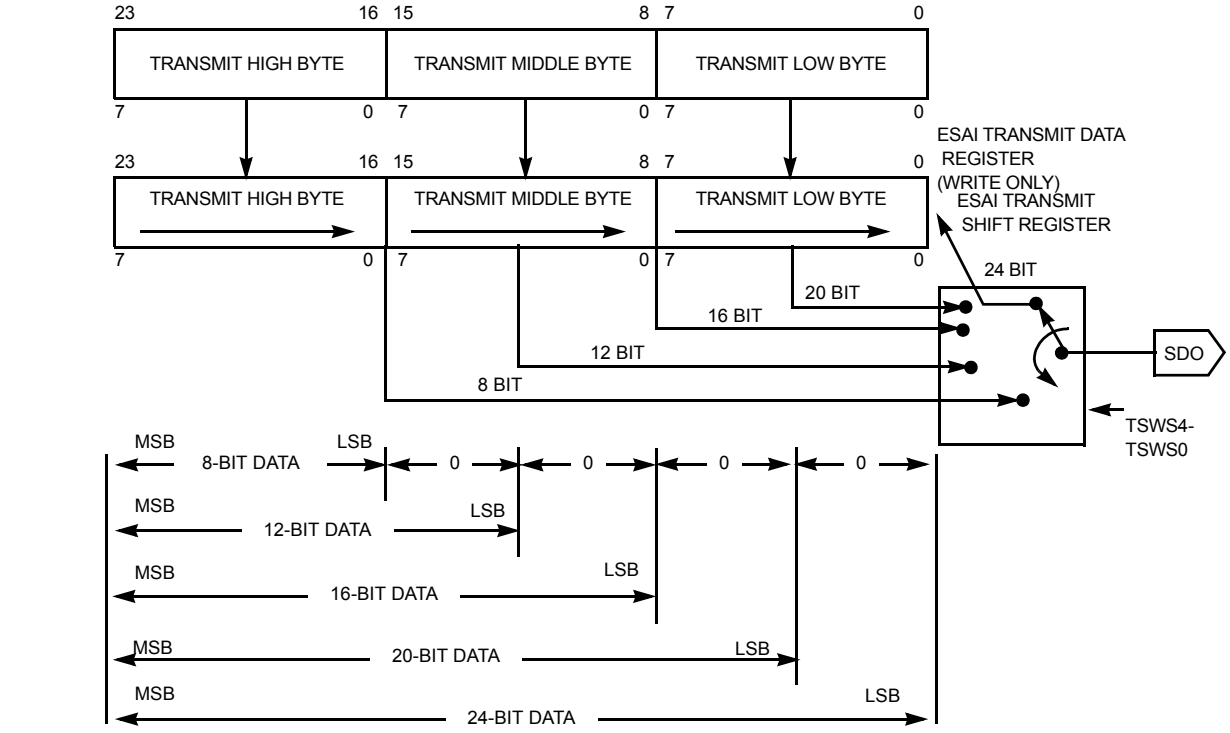


Figure 6-13 ESAI Data Path Programming Model ([R/T]SHFD=0)



(a) Receive Registers

NOTES:
1. Data is received LSB first if RSHFD=1.
2. 24-bit fractional format (ALC=0).
3. 32-bit mode is not shown.



(b) Transmit Registers

NOTES:
1. Data is sent LSB first if TSHFD=1.
2. 24-bit fractional format (ALC=0).
3. 32-bit mode is not shown.
4. Data word is left aligned (TWA=0,PADC=1).

Figure 6-14 ESAI Data Path Programming Model ([R/T]SHFD=1)

6.3.7 ESAI Receive Shift Registers

The receive shift registers (see Figure 6-13 and Figure 6-14) receive the incoming data from the serial receive data pins. Data is shifted in by the selected (internal/external) bit clock when the associated frame sync I/O is asserted. Data is assumed to be received MSB first if RSHFD=0 and LSB first if RSHFD=1. Data is transferred to the ESAI receive data registers after 8, 12, 16, 20, 24, or 32 serial clock cycles were counted, depending on the slot length control bits in the RCR register.

6.3.8 ESAI Receive Data Registers (RX3, RX2, RX1, RX0)

RX3, RX2, RX1 and RX0 are 24-bit read-only registers that accept data from the receive shift registers when they become full (see Figure 6-13 and Figure 6-14). The data occupies the most significant portion of the receive data registers, according to the ALC control bit setting. The unused bits (least significant portion, and 8 most significant bits when ALC=1) read as zeros. The DSP is interrupted whenever RXx becomes full if the associated interrupt is enabled.

6.3.9 ESAI Transmit Shift Registers

The transmit shift registers contain the data being transmitted (see Figure 6-13 and Figure 6-14). Data is shifted out to the serial transmit data pins by the selected (internal/external) bit clock when the associated frame sync I/O is asserted. The number of bits shifted out before the shift registers are considered empty and may be written to again can be 8, 12, 16, 20, 24 or 32 bits (determined by the slot length control bits in the TCR register). Data is shifted out of these registers MSB first if TSHFD=0 and LSB first if TSHFD=1.

6.3.10 ESAI Transmit Data Registers (TX5, TX4, TX3, TX2,TX1,TX0)

TX5, TX4, TX3, TX2, TX1 and TX0 are 24-bit write-only registers. Data to be transmitted is written into these registers and is automatically transferred to the transmit shift registers (see Figure 6-13 and Figure 6-14). The data written (8, 12, 16, 20 or 24 bits) should occupy the most significant portion of the TXx according to the ALC control bit setting. The unused bits (least significant portion, and the 8 most significant bits when ALC=1) of the TXx are don’t care bits. The DSP is interrupted whenever the TXx becomes empty if the transmit data register empty interrupt has been enabled.

6.3.11 ESAI Time Slot Register (TSR)

The write-only Time Slot Register (TSR) is effectively a null data register that is used when the data is not to be transmitted in the available transmit time slot. The transmit data pins of all the enabled transmitters are in the high-impedance state for the respective time slot where TSR has been written. The Transmitter External Buffer Enable pin (FSR pin when SYN=1, TEBE=1, RFSD=1) disables the external buffers during the slot when the TSR register has been written.

6.3.12 Transmit Slot Mask Registers (TSMA, TSMB)

The Transmit Slot Mask Registers (TSMA and TSMB) are two read/write registers used by the transmitters in network mode to determine for each slot whether to transmit a data word and generate a

transmitter empty condition (TDE=1), or to tri-state the transmitter data pins. TSMA and TSMB should each be considered as containing half a 32-bit register TSM. See Figure 6-15 and Figure 6-16. Bit number N in TSM (TS**) is the enable/disable control bit for transmission in slot number N.

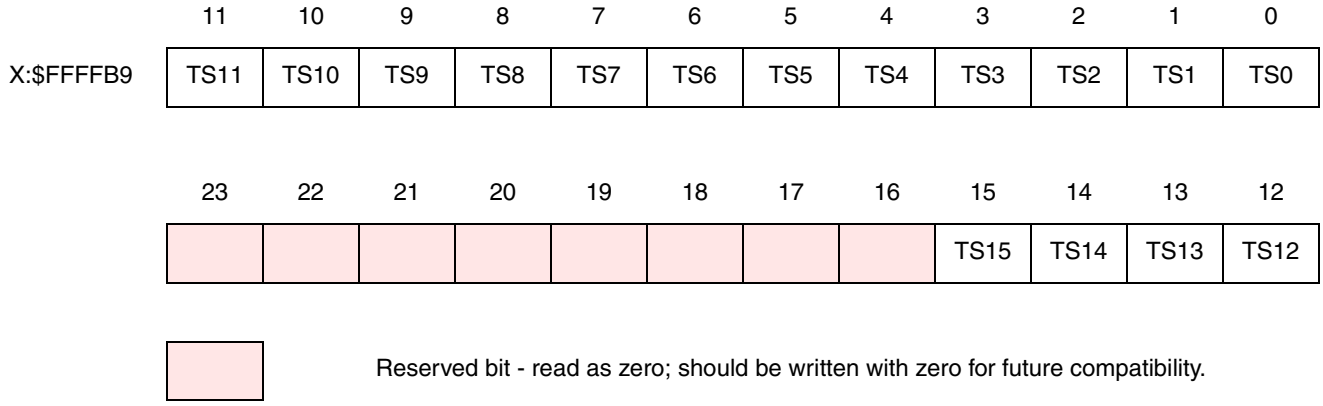


Figure 6-15 TSMA Register

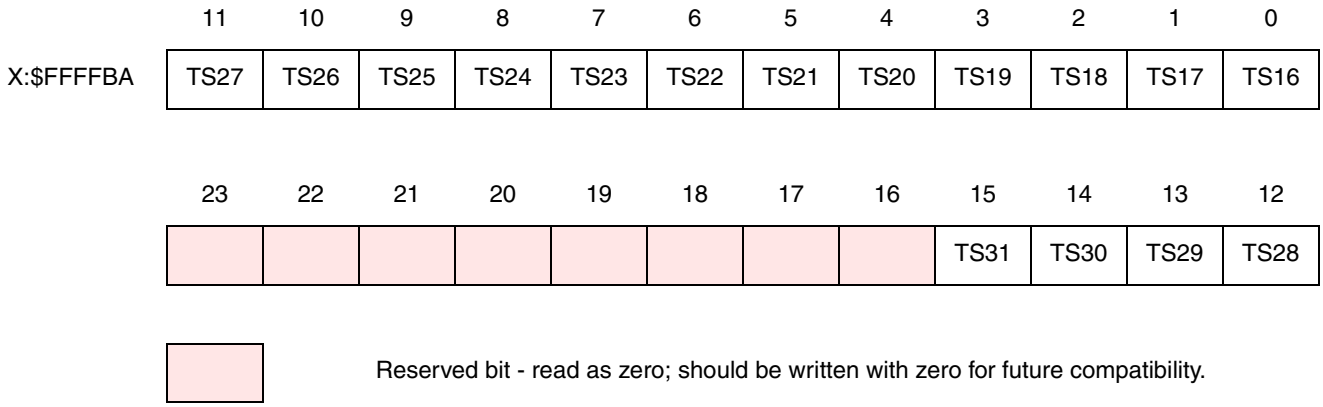


Figure 6-16 TSMB Register

When bit number N in TSM is cleared, all the transmit data pins of the enabled transmitters are tri-stated during transmit time slot number N. The data is still transferred from the transmit data registers to the transmit shift registers but neither the TDE nor the TUE flags are set. This means that during a disabled slot, no transmitter empty interrupt is generated. The DSP is interrupted only for enabled slots. Data that is written to the transmit data registers when servicing this request is transmitted in the next enabled transmit time slot.

When bit number N in TSM register is set, the transmit sequence is as usual: data is transferred from the TX registers to the shift registers, transmitted during slot number N, and the TDE flag is set.

Using the slot mask in TSM does not conflict with using TSR. Even if a slot is enabled in TSM, the user may chose to write to TSR instead of writing to the transmit data registers TXx. This causes all the transmit data pins of the enabled transmitters to be tri-stated during the next slot.

Data written to the TSM affects the next frame transmission. The frame being transmitted is not affected by this data and would comply to the last TSM setting. Data read from TSM returns the last written data.

After hardware or software reset, the TSM register is preset to $FFFFFFFF, which means that all 32 possible slots are enabled for data transmission.

NOTE

When operating in normal mode, bit 0 of the mask register must be set, otherwise no output is generated.

6.3.13 Receive Slot Mask Registers (RSMA, RSMB)

The Receive Slot Mask Registers (RSMA and RSMB) are two read/write registers used by the receiver in network mode to determine for each slot whether to receive a data word and generate a receiver full condition (RDF=1), or to ignore the received data. RSMA and RSMB should be considered as each containing half of a 32-bit register RSM. See Figure 6-17 and Figure 6-18. Bit number N in RSM (RS**) is an enable/disable control bit for receiving data in slot number N.

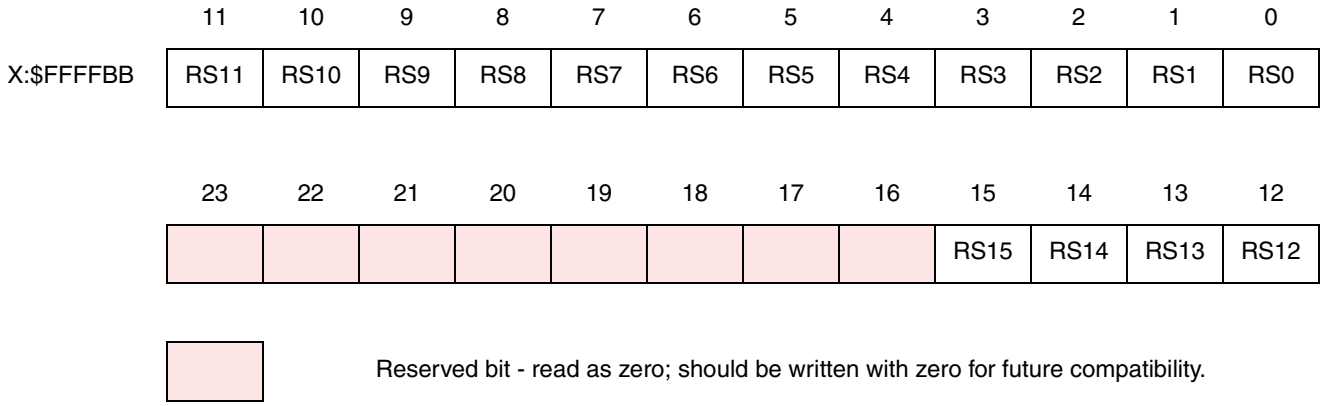


Figure 6-17 RSMA Register

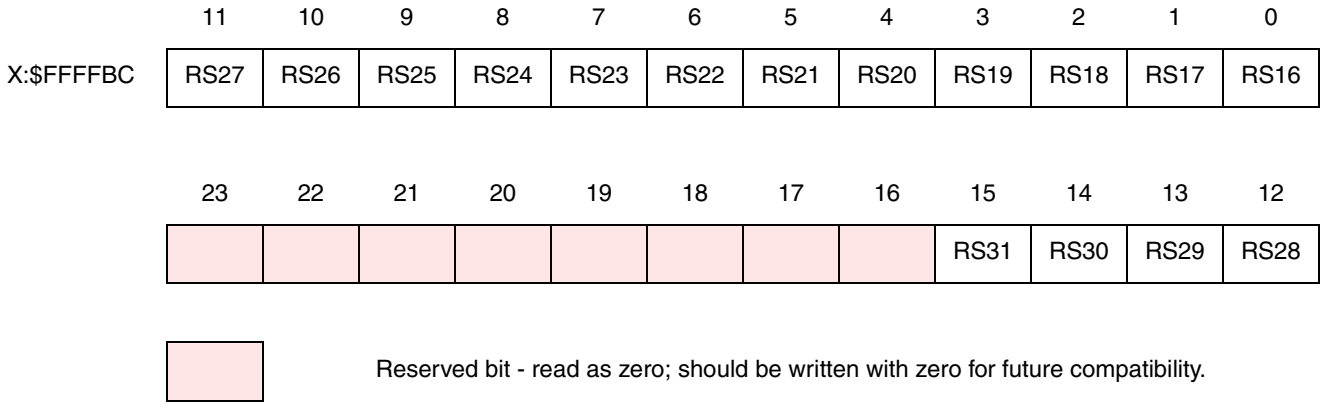


Figure 6-18 RSMB Register

When bit number N in the RSM register is cleared, the data from the enabled receivers input pins are shifted into their receive shift registers during slot number N. The data is not transferred from the receive shift registers to the receive data registers, and neither the RDF nor the ROE flags are set. This means that during a disabled slot, no receiver full interrupt is generated. The DSP is interrupted only for enabled slots.

When bit number N in the RSM is set, the receive sequence is as usual: data which is shifted into the enabled receivers shift registers is transferred to the receive data registers and the RDF flag is set.

Data written to the RSM affects the next received frame. The frame being received is not affected by this data and would comply to the last RSM setting. Data read from RSM returns the last written data.

After hardware or software reset, the RSM register is preset to $FFFFFFFF, which means that all 32 possible slots are enabled for data reception.

NOTE

When operating in normal mode, bit 0 of the mask register must be set to one, otherwise no input is received.

6.4 Operating Modes

ESAI operating mode are selected by the ESAI control registers (TCCR, TCR, RCCR, RCR and SAICR). The main operating mode are described in the following paragraphs.

6.4.1 ESAI After Reset

Hardware or software reset clears the port control register bits and the port direction control register bits, which configure all ESAI I/O pins as disconnected. The ESAI is in the individual reset state while all ESAI pins are programmed as GPIO or disconnected, and is active only if at least one of the ESAI I/O pins is programmed as an ESAI pin.

6.4.2 ESAI Initialization

The correct way to initialize the ESAI is as follows:

1. Hardware, software, ESAI individual, or STOP reset.
2. Program ESAI control and time slot registers.
3. Write data to all the enabled transmitters.
4. Configure at least one pin as ESAI pin.

During program execution, all ESAI pins may be defined as GPIO or disconnected, causing the ESAI to stop serial activity and enter the individual reset state. All status bits of the interface are set to their reset state; however, the control bits are not affected. This procedure allows the DSP programmer to reset the ESAI separately from the other internal peripherals. During individual reset, internal DMA accesses to the data registers of the ESAI are not valid and data read is undefined.

The DSP programmer must use an individual ESAI reset when changing the ESAI control registers (except for TEIE, REIE, TLIE, RLIE, TIE, RIE, TE0-TE5, RE0-RE3) to ensure proper operation of the interface.

NOTE

If the ESAI receiver section is already operating with some of the receivers, enabling additional receivers on the fly (i.e. without first putting the ESAI receiver in the personal reset state) by setting their REx control bits will result in erroneous data being received as the first data word for the newly enabled receivers.

6.4.3 ESAI Interrupt Requests

The ESAI can generate eight different interrupt requests (ordered from the highest to the lowest priority)**:**

1. ESAI Receive Data with Exception Status.
   Occurs when the receive exception interrupt is enabled (REIE=1 in the RCR register), at least one of the enabled receive data registers is full (RDF=1), and a receiver overrun error has occurred (ROE=1 in the SAISR register). ROE is cleared by first reading the SAISR and then reading all the enabled receive data registers.
2. ESAI Receive Even Data
   Occurs when the receive even slot data interrupt is enabled (REDIE=1), at least one of the enabled receive data registers is full (RDF=1), the data is from an even slot (REDF=1), and no exception has occurred (ROE=0 or REIE=0).
   Reading all enabled receiver data registers clears RDF and REDF.
3. ESAI Receive Data
   Occurs when the receive interrupt is enabled (RIE=1), at least one of the enabled receive data registers is full (RDF=1), no exception has occurred (ROE=0 or REIE=0), and no even slot interrupt has occurred (REDF=0 or REDIE=0).
   Reading all enabled receiver data registers clears RDF.
4. ESAI Receive Last Slot Interrupt
   Occurs, if enabled (RLIE=1), after the last slot of the frame ended (in network mode only) regardless of the receive mask register setting. The receive last slot interrupt may be used for resetting the receive mask slot register, reconfiguring the DMA channels and reassigning data memory pointers. Using the receive last slot interrupt guarantees that the previous frame was serviced with the previous setting and the new frame is serviced with the new setting without synchronization problems. Note that the maximum receive last slot interrupt service time should not exceed N-1 ESAI bits service time (where N is the number of bits in a slot).
5. ESAI Transmit Data with Exception Status
   Occurs when the transmit exception interrupt is enabled (TEIE=1), at least one transmit data register of the enabled transmitters is empty (TDE=1), and a transmitter underrun error has occurred (TUE=1). TUE is cleared by first reading the SAISR and then writing to all the enabled transmit data registers, or to the TSR register.
6. ESAI Transmit Last Slot Interrupt
   Occurs, if enabled (TLIE=1), at the start of the last slot of the frame in network mode regardless of the transmit mask register setting. The transmit last slot interrupt may be used for resetting the transmit mask slot register, reconfiguring the DMA channels and reassigning data memory pointers. Using the transmit last slot interrupt guarantees that the previous frame was serviced with

the previous setting and the new frame is serviced with the new setting without synchronization problems. Note that the maximum transmit last slot interrupt service time should not exceed N-1 ESAI bits service time (where N is the number of bits in a slot).

7. ESAI Transmit Even Data
   Occurs when the transmit even slot data interrupt is enabled (TEDIE=1), at least one of the enabled transmit data registers is empty (TDE=1), the slot is an even slot (TEDE=1), and no exception has occurred (TUE=0 or TEIE=0).
   Writing to all the TX registers of the enabled transmitters or to TSR clears this interrupt request.
8. ESAI Transmit Data
   Occurs when the transmit interrupt is enabled (TIE=1), at least one of the enabled transmit data registers is empty (TDE=1), no exception has occurred (TUE=0 or TEIE=0), and no even slot interrupt has occurred (TEDE=0 or TEDIE=0).
   Writing to all the TX registers of the enabled transmitters, or to the TSR clears this interrupt request.

6.4.4 Operating Modes – Normal, Network, and On-Demand

The ESAI has three basic operating modes and many data/operation formats.

6.4.4.1 Normal/Network/On-Demand Mode Selection

Selecting between the normal mode and network mode is accomplished by clearing or setting the TMOD0-TMOD1 bits in the TCR register for the transmitter section, and in the RMOD0-RMOD1 bits in the RCR register for the receiver section.

For normal mode, the ESAI functions with one data word of I/O per frame (per enabled transmitter or receiver). The normal mode is typically used to transfer data to/from a single device.

For the network mode, 2 to 32 time slots per frame may be selected. During each frame, 0 to 32 data words of I/O may be received/transmitted. In either case, the transfers are periodic. The frame sync signal indicates the first time slot in the frame. Network mode is typically used in time division multiplexed (TDM) networks of codecs, DSPs with multiple words per frame, or multi-channel devices.

Selecting the network mode and setting the frame rate divider to zero (DC=00000) selects the on-demand mode. This special case does not generate a periodic frame sync. A frame sync pulse is generated only when data is available to transmit. The on-demand mode requires that the transmit frame sync be internal (output) and the receive frame sync be external (input). Therefore, for simplex operation, the synchronous mode could be used; however, for full-duplex operation, the asynchronous mode must be used. Data transmission that is data driven is enabled by writing data into each TX. Although the ESAI is double buffered, only one word can be written to each TX, even if the transmit shift register is empty. The receive and transmit interrupts function as usual using TDE and RDF; however, transmit underruns are impossible for on-demand transmission and are disabled.

6.4.4.2 Synchronous/Asynchronous Operating Modes

The transmit and receive sections of the ESAI may be synchronous or asynchronous – i.e., the transmitter and receiver sections may use common clock and synchronization signals (synchronous operating mode),

or they may have their own separate clock and sync signals (asynchronous operating mode). The SYN bit in the SAICR register selects synchronous or asynchronous operation. Since the ESAI is designed to operate either synchronously or asynchronously, separate receive and transmit interrupts are provided.

When SYN is cleared, the ESAI transmitter and receiver clocks and frame sync sources are independent. If SYN is set, the ESAI transmitter and receiver clocks and frame sync come from the transmitter section (either external or internal sources).

Data clock and frame sync signals can be generated internally by the DSP or may be obtained from external sources. If internally generated, the ESAI clock generator is used to derive high frequency clock, bit clock and frame sync signals from the DSP internal system clock.

6.4.4.3 Frame Sync Selection

The frame sync can be either a bit-long or word-long signal. The transmitter frame format is defined by the TFSL bit in the TCR register. The receiver frame format is defined by the RFSL bit in the RCR register.

1. In the word-long frame sync format, the frame sync signal is asserted during the entire word data transfer period. This frame sync length is compatible with Freescale codecs, SPI serial peripherals, serial A/D and D/A converters, shift registers, and telecommunication PCM serial I/O.
2. In the bit-long frame sync format, the frame sync signal is asserted for one bit clock immediately before the data transfer period. This frame sync length is compatible with Intel and National components, codecs, and telecommunication PCM serial I/O.

The relative timing of the word length frame sync as referred to the data word is specified by the TFSR bit in the TCR register for the transmitter section, and by the RFSR bit in the RCR register for the receive section. The word length frame sync may be generated (or expected) with the first bit of the data word, or with the last bit of the previous word. TFSR and RFSR are ignored when a bit length frame sync is selected.

Polarity of the frame sync signal may be defined as positive (asserted high) or negative (asserted low). The TFSP bit in the TCCR register specifies the polarity of the frame sync for the transmitter section. The RFSP bit in the RCCR register specifies the polarity of the frame sync for the receiver section.

The ESAI receiver looks for a receive frame sync leading edge (trailing edge if RFSP is set) only when the previous frame is completed. If the frame sync goes high before the frame is completed (or before the last bit of the frame is received in the case of a bit frame sync or a word length frame sync with RFSR set), the current frame sync is not recognized, and the receiver is internally disabled until the next frame sync. Frames do not have to be adjacent – i.e., a new frame sync does not have to immediately follow the previous frame. Gaps of arbitrary periods can occur between frames. Enabled transmitters are tri-stated during these gaps.

When operating in the synchronous mode (SYN=1), all clocks including the frame sync are generated by the transmitter section.

6.4.4.4 Shift Direction Selection

Some data formats, such as those used by codecs, specify MSB first while other data formats, such as the AES-EBU digital audio interface, specify LSB first. The MSB/LSB first selection is made by

programming RSHFD bit in the RCR register for the receiver section, and by programming the TSHFD bit in the TCR register for the transmitter section.

6.4.5 Serial I/O Flags

Three ESAI pins (FSR, SCKR and HCKR) are available as serial I/O flags when the ESAI is operating in the synchronous mode (SYN=1). Their operation is controlled by RCKD, RFSD, TEBE bits in the RCR, RCCR and SAICR registers.The output data bits (OF2, OF1 and OF0) and the input data bits (IF2, IF1 and IF0) are double buffered to/from the HCKR, FSR and SCKR pins. Double buffering the flags keeps them in sync with the TX and RX data lines.

Each flag can be separately programmed. Flag 0 (SCKR pin) direction is selected by RCKD, RCKD=1 for output and RCKD=0 for input. Flag 1 (FSR pin) is enabled when the pin is not configured as external transmitter buffer enable (TEBE=0) and its direction is selected by RFSD, RFSD=1 for output and RFSD=0 for input. Flag 2 (HCKR pin) direction is selected by RHCKD, RHCKD=1 for output and RHCKD=0 for input.

When programmed as input flags, the SCKR, FSR and HCKR logic values, respectively, are latched at the same time as the first bit of the receive data word is sampled. Because the input was latched, the signal on the input flag pin (SCKR, FSR or HCKR) can change without affecting the input flag until the first bit of the next receive data word. When the received data words are transferred to the receive data registers, the input flag latched values are then transferred to the IF0, IF1 and IF2 bits in the SAISR register, where they may be read by software.

When programmed as output flags, the SCKR, FSR and HCKR logic values are driven by the contents of the OF0, OF1 and OF2 bits in the SAICR register respectively, and are driven when the transmit data registers are transferred to the transmit shift registers. The value on SCKR, FSR and HCKR is stable from the time the first bit of the transmit data word is transmitted until the first bit of the next transmit data word is transmitted. Software may change the OF0-OF2 values thus controlling the SCKR, FSR and HCKR pin values for each transmitted word. The normal sequence for setting output flags when transmitting data is as follows: wait for TDE (transmitter empty) to be set, first write the flags, and then write the transmit data to the transmit registers. OF0, OF1 and OF2 are double buffered so that the flag states appear on the pins when the transmit data is transferred to the transmit shift register (i.e., the flags are synchronous with the data).

6.5 GPIO - Pins and Registers

The GPIO functionality of the ESAI port is controlled by three registers: Port C control register (PCRC), Port C direction register (PRRC) and Port C data register (PDRC).

6.5.1 Port C Control Register (PCRC)

The read/write 24-bit Port C Control Register (PCRC) in conjunction with the Port C Direction Register (PRRC) controls the functionality of the ESAI GPIO pins. Each of the PC(11:0) bits controls the functionality of the corresponding port pin. See Table 6-12 for the port pin configurations. Hardware and software reset clear all PCRC bits.

6.5.2 Port C Direction Register (PRRC)

The read/write 24-bit Port C Direction Register (PRRC) in conjunction with the Port C Control Register (PCRC) controls the functionality of the ESAI GPIO pins. Table 6-12 describes the port pin configurations. Hardware and software reset clear all PRRC bits.

Table 6-12 PCRC and PRRC Bits Functionality

PDC[i]	PC[i]	Port Pin[i] Function
0	0	disconnected
0	1	GPIO input
1	0	GPIO output
1	1	ESAI

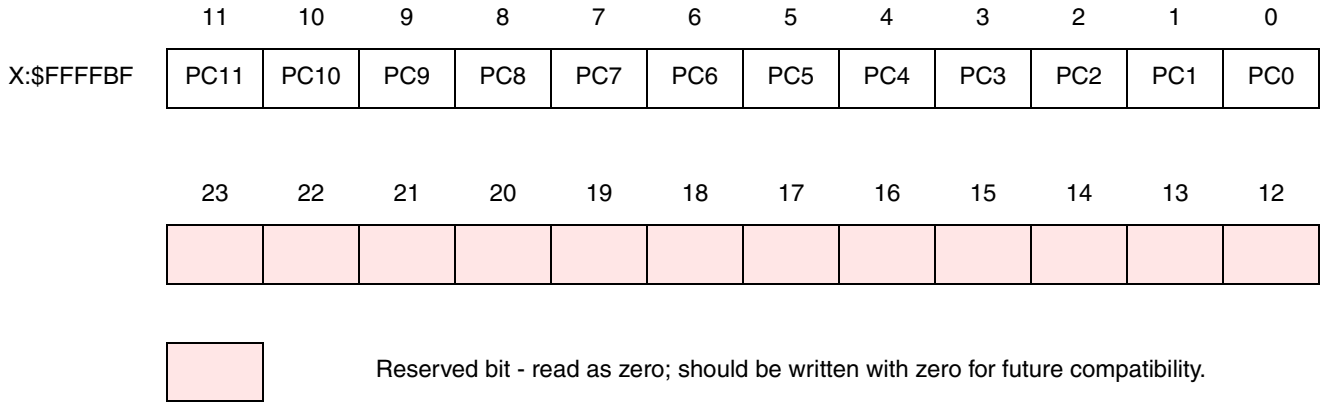


Figure 6-19 PCRC Register

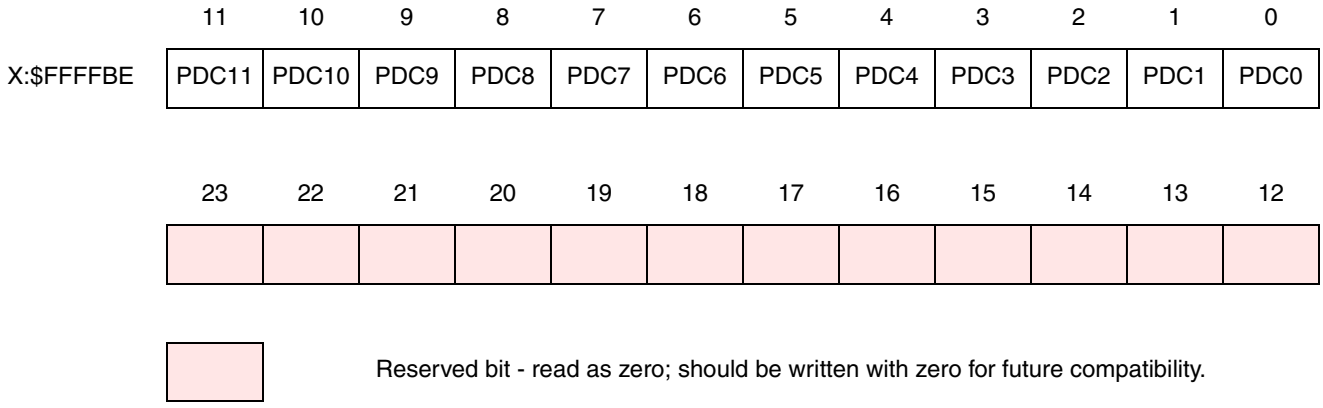


Figure 6-20 PRRC Register

6.5.3 Port C Data register (PDRC)

The read/write 24-bit Port C Data Register (see Figure 6-21) is used to read or write data to/from ESAI GPIO pins. Bits PD(11:0) are used to read or write data from/to the corresponding port pins if they are

configured as GPIO. If a port pin [i] is configured as a GPIO input, then the corresponding PD[i] bit reflects the value present on this pin. If a port pin [i] is configured as a GPIO output, then the value written into the corresponding PD[i] bit is reflected on this pin. If a port pin [i] is configured as disconnected, the corresponding PD[i] bit is not reset and contains undefined data.

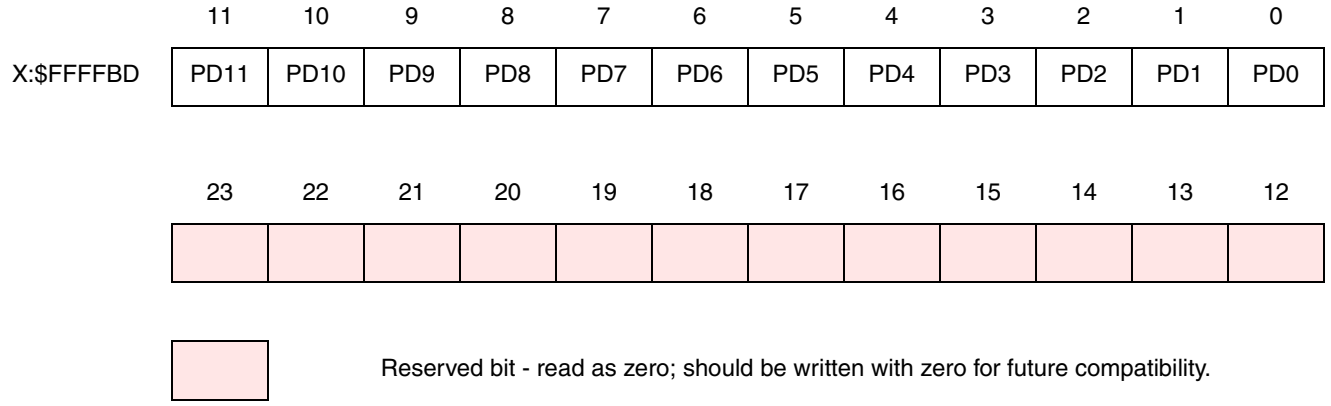


Figure 6-21 PDRC Register

6.6 ESAI Initialization Examples

6.6.1 Initializing the ESAI Using Individual Reset

- The ESAI should be in its individual reset state (PCRC = $000 and PRRC = $000). In the individual reset state, both the transmitter and receiver sections of the ESAI are simultaneously reset. The TPR bit in the TCR register may be used to reset just the transmitter section. The RPR bit in the RCR register may be used to reset just the receiver section.
- Configure the control registers (TCCR, TCR, RCCR, RCR) according to the operating mode, but do not enable transmitters (TE5–TE0 = $0) or receivers (RE3–RE0 = $0). It is possible to set the interrupt enable bits which are in use during the operation (no interrupt occurs).
- Enable the ESAI by setting the PCRC register and PRRC register bits according to pins which are in use during operation.
- Write the first data to be transmitted to the transmitters which are in use during operation. This step is needed even if DMA is used to service the transmitters.
- Enable the transmitters and receivers.
- From now on ESAI can be serviced either by polling, interrupts, or DMA.

Operation proceeds as follows:

- For internally generated clock and frame sync, these signals are active immediately after ESAI is enabled (step 3 above).
- Data is received only when one of the receive enable (REx) bits is set and after the occurrence of frame sync signal (either internally or externally generated).
- Data is transmitted only when the transmitter enable (TEx) bit is set and after the occurrence of frame sync signal (either internally or externally generated). The transmitter outputs remain tri-stated after TEx bit is set until the frame sync occurs.

6.6.2 Initializing Just the ESAI Transmitter Section

- It is assumed that the ESAI is operational; that is, at least one pin is defined as an ESAI pin.
- The transmitter section should be in its personal reset state (TPR = 1).
- Configure the control registers TCCR and TCR according to the operating mode, making sure to clear the transmitter enable bits (TE0 - TE5). TPR must remain set.
- Take the transmitter section out of the personal reset state by clearing TPR.
- Write first data to the transmitters which will be used during operation. This step is needed even if DMA is used to service the transmitters.
- Enable the transmitters by setting their TE bits.
- Data is transmitted only when the transmitter enable (TEx) bit is set and after the occurrence of frame sync signal (either internally or externally generated). The transmitter outputs remain tri-stated after TEx bit is set until the frame sync occurs.
- From now on the transmitters are operating and can be serviced either by polling, interrupts, or DMA.

6.6.3 Initializing Just the ESAI Receiver Section

- It is assumed that the ESAI is operational; that is, at least one pin is defined as an ESAI pin.
- The receiver section should be in its personal reset state (RPR = 1).
- Configure the control registers RCCR and RCR according to the operating mode, making sure to clear the receiver enable bits (RE0 - RE3). RPR must remain set.
- Take the receiver section out of the personal reset state by clearing RPR.
- Enable the receivers by setting their RE bits.
- From now on the receivers are operating and can be serviced either by polling, interrupts, or DMA.

NOTES

7 Serial Host Interface

7.1 Introduction

The Serial Host Interface (SHI) is a serial I/O interface that provides a path for communication and program/coefficient data transfers between the DSP and an external host processor. The SHI can also communicate with other serial peripheral devices. The SHI can interface directly to either of two well-known and widely used synchronous serial buses: the Freescale Serial Peripheral Interface (SPI) bus and the Philips Inter-Integrated-circuit Control (I^2C) bus. The SHI supports either the SPI or I^2C bus protocol, as required, from a slave or a single-master device. To minimize DSP overhead, the SHI supports single-, double-, and triple-byte data transfers. The SHI has a 10-word receive FIFO that permits receiving up to 30 bytes before generating a receive interrupt, reducing the overhead for data reception.

When configured in the SPI mode, the SHI can:

- Identify its slave selection (in Slave mode)
- Simultaneously transmit (shift out) and receive (shift in) serial data
- Directly operate with 8-, 16- and 24-bit words
- Generate vectored interrupts, separately for receive and transmit events, and update status bits
- Generate a separate vectored interrupt in the event of a receive exception
- Generate a separate vectored interrupt in the event of a bus-error exception
- Optionally trigger DMA interrupts to service the transmit and receive events
- Generate the serial clock signal (in Master mode)

When configured in the I^2C mode, the SHI can:

- Detect/generate start and stop events
- Identify its slave (ID) address (in Slave mode)
- Identify the transfer direction (receive/transmit)
- Transfer data byte-wise according to the SCL clock line
- Generate ACK signal following a byte receive
- Inspect ACK signal following a byte transmit
- Directly operate with 8-, 16- and 24-bit words
- Generate vectored interrupts separately for receive and transmit events and update status bits
- Generate a separate vectored interrupt in the event of a receive exception
- Generate a separate vectored interrupt in the event of a bus error exception
- Optionally trigger DMA interrupts to service the transmit and receive events
- Generate the clock signal (in Master mode)

7.2 Serial Host Interface Internal Architecture

The DSP views the SHI as a memory-mapped peripheral in the X data memory space. The DSP may access the SHI as a normal memory-mapped peripheral using standard polling or interrupt programming techniques and DMA transfers. Memory mapping allows DSP communication with the SHI registers to be accomplished using standard instructions and addressing modes. In addition, the MOVEP instruction allows interface-to-memory and memory-to-interface data transfers without going through an intermediate register. The DMA controller may be used to service the receive or transmit data path. The single master configuration allows the DSP to directly connect to dumb peripheral devices. For that purpose, a programmable baud-rate generator is included to generate the clock signal for serial transfers. The host side invokes the SHI, for communication and data transfer with the DSP, through a shift register that may be accessed serially using either the I^2C or the SPI bus protocols. Figure 7-1 shows the SHI block diagram.

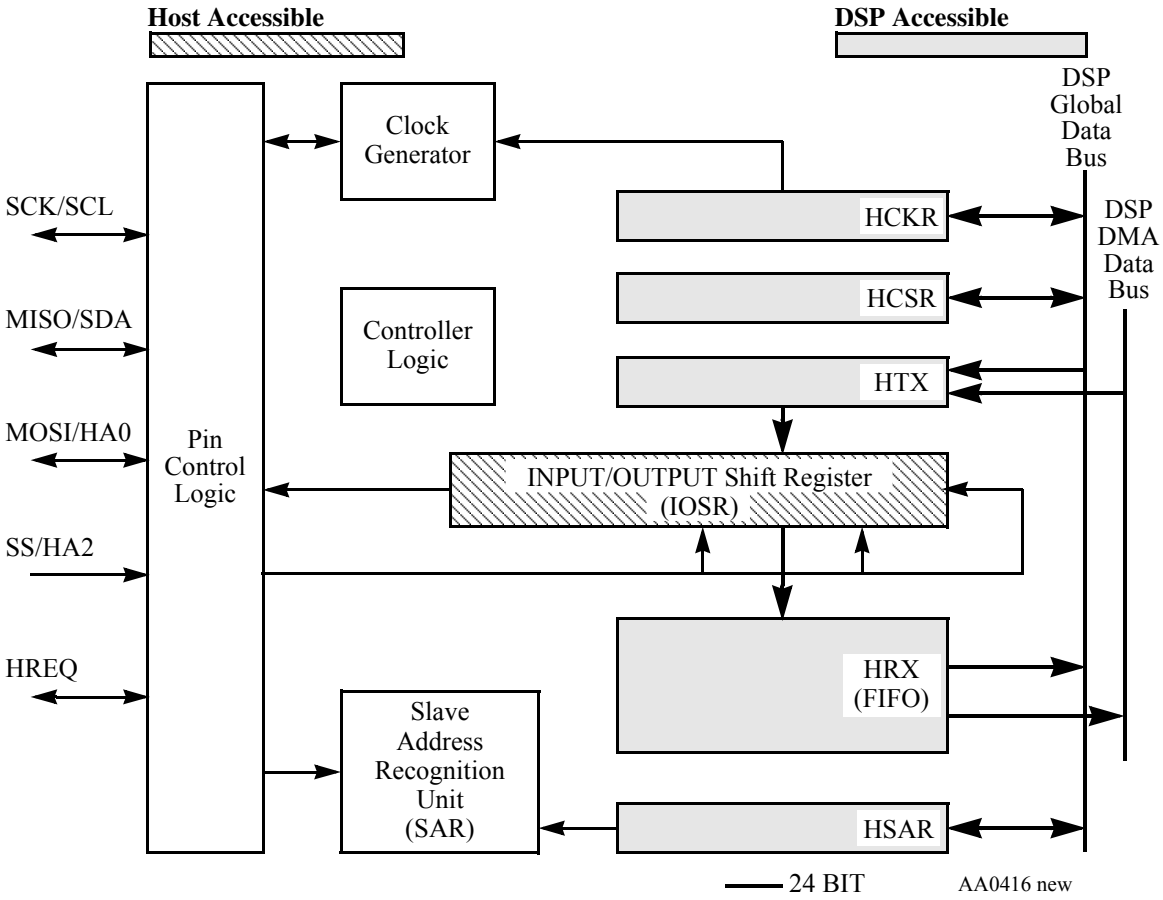


Figure 7-1 Serial Host Interface Block Diagram

7.3 SHI Clock Generator

The SHI clock generator generates the serial clock to the SHI if the interface operates in the Master mode. The clock generator is disabled if the interface operates in the Slave mode, except in I^2C mode when the HCKFR bit is set in the HCKR register. When the SHI operates in the Slave mode, the clock is external and is input to the SHI (HMST = 0). Figure 7-2 illustrates the internal clock path connections. It is the user's responsibility to select the proper clock rate within the range as defined in the I^2C and SPI bus specifications.

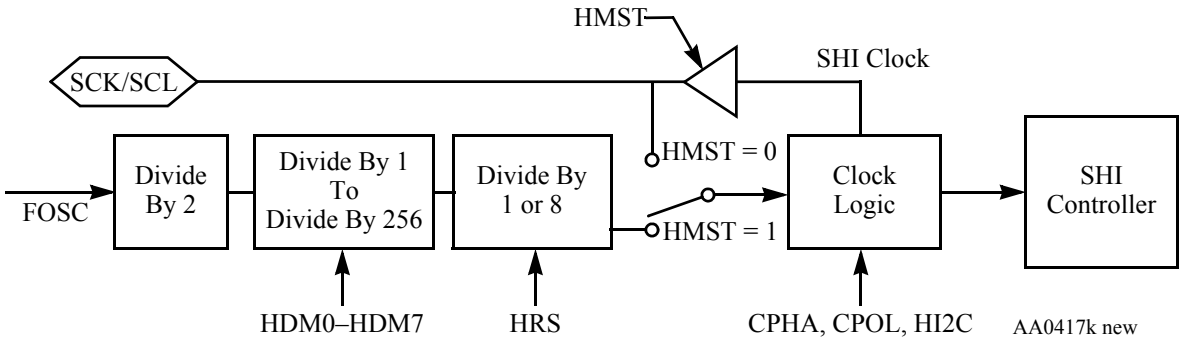


Figure 7-2 SHI Clock Generator

7.4 Serial Host Interface Programming Model

The Serial Host Interface programming model is divided in two parts:

- **Host side**—see Figure 7-3 below and Section 7.4.1, "SHI Input/Output Shift Register (IOSR)—Host Side"
- **DSP side**—see Figure 7-4 and Section 7.4.2, "SHI Host Transmit Data Register (HTX)—DSP Side" through Section 7.4.6, "SHI Control/Status Register (HCSR)—DSP Side" for detailed information.

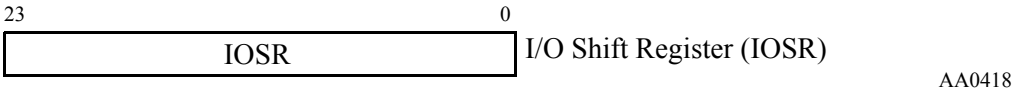


Figure 7-3 SHI Programming Model—Host Side

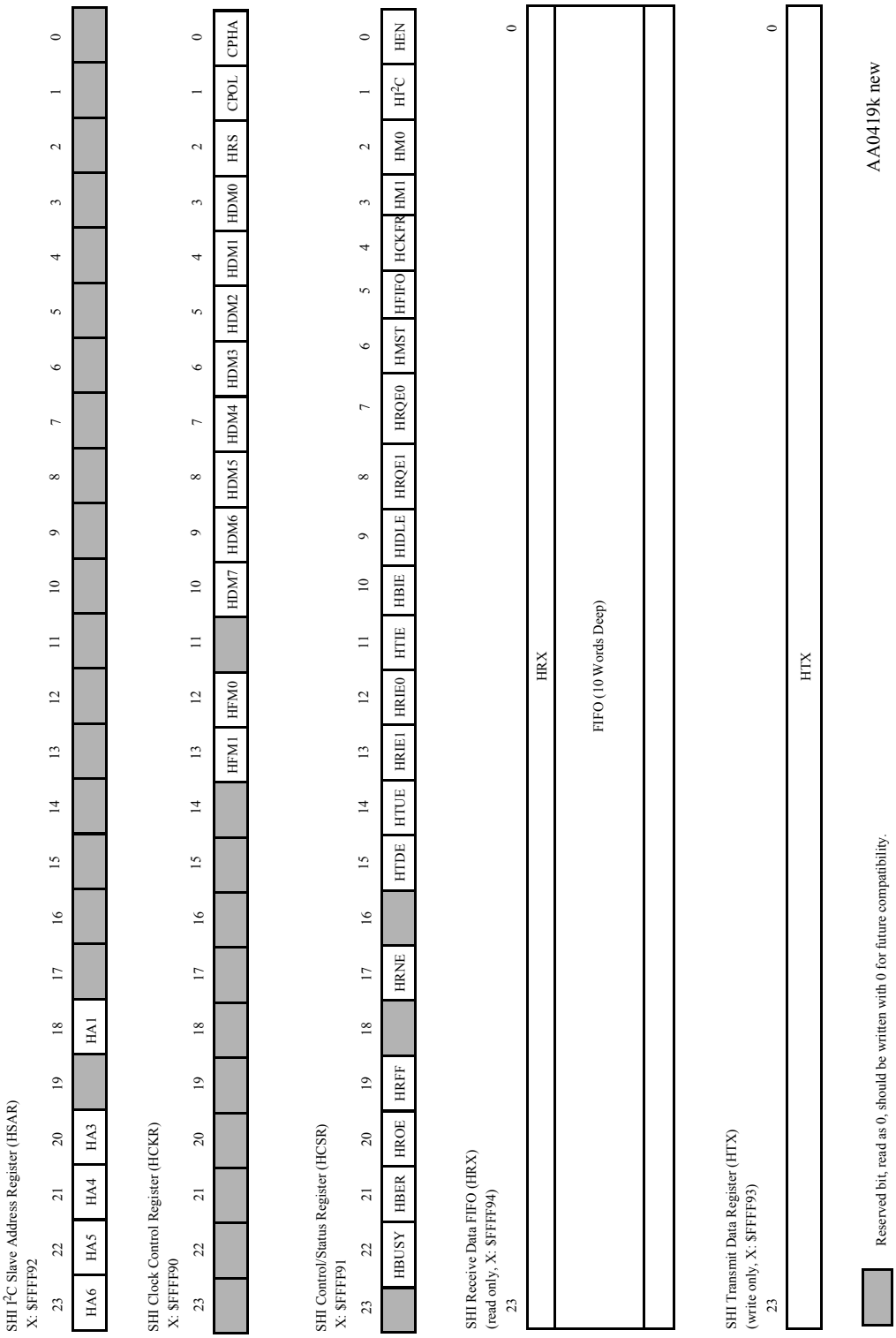


Figure 7-4 SHI Programming Model—DSP Side

The interrupt vector table for the Serial Host Interface is shown in Table 7-1 and the exceptions generated by the SHI are prioritized as shown in Table 7-2.

Table 7-1 SHI Interrupt Vectors

Program Address	Interrupt Source
VBA:$40	SHI Transmit Data
VBA:$42	SHI Transmit Underrun Error
VBA:$44	SHI Receive FIFO Not Empty
VBA:$48	SHI Receive FIFO Full
VBA:$4A	SHI Receive Overrun Error
VBA:$4C	SHI Bus Error

Table 7-2 SHI Internal Interrupt Priorities

Priority	Interrupt
Highest	SHI Bus Error
	SHI Receive Overrun Error
	SHI Transmit Underrun Error
	SHI Receive FIFO Full
	SHI Transmit Data
Lowest	SHI Receive FIFO Not Empty

7.4.1 SHI Input/Output Shift Register (IOSR)—Host Side

The variable length Input/Output Shift Register (IOSR) can be viewed as a serial-to-parallel and parallel-to-serial buffer in the SHI. The IOSR is involved with every data transfer in both directions (read and write). In compliance with the I^2C and SPI bus protocols, data is shifted in and out MSB first. In single-byte data transfer modes, the most significant byte of the IOSR is used as the shift register. In 16-bit data transfer modes, the two most significant bytes become the shift register. In 24-bit transfer modes, the shift register uses all three bytes of the IOSR (see Figure 7-5).

NOTE

The IOSR cannot be accessed directly either by the host processor or by the DSP. It is fully controlled by the SHI controller logic.

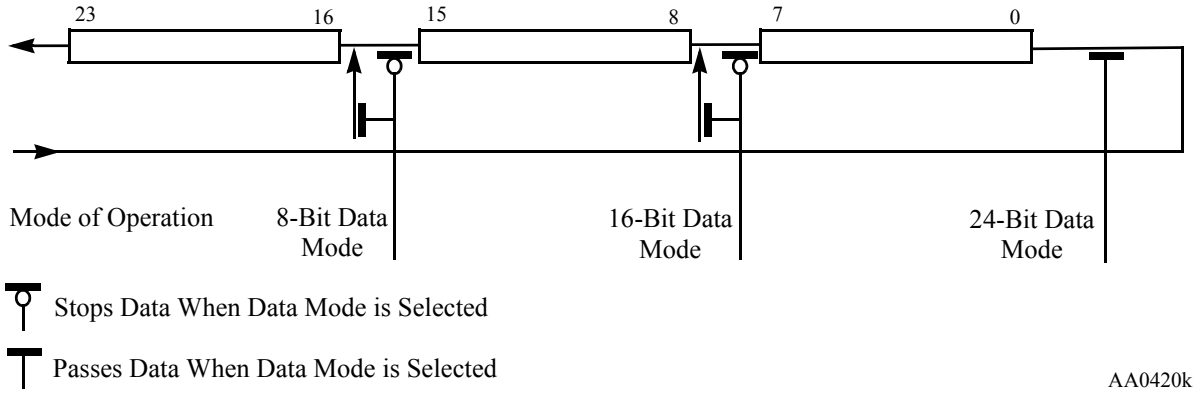


Figure 7-5 SHI I/O Shift Register (IOSR)

7.4.2 SHI Host Transmit Data Register (HTX)—DSP Side

The Host Transmit data register (HTX) is used for DSP-to-Host data transfers. The HTX register is 24 bits wide. Writing to the HTX register by DSP core instructions or by DMA transfers clears the HTDE flag. The DSP may program the HTIE bit to cause a Host transmit data interrupt when HTDE is set (see Section 7.4.6.11, "HCSR Transmit-Interrupt Enable (HTIE)—Bit 11"). Data should not be written to the HTX until HTDE is set in order to prevent overwriting the previous data. HTX is reset to the empty state when in Stop mode and during hardware reset, software reset, and individual reset.

In the single-byte data transfer mode the most significant byte of the HTX is transmitted; in the double-byte mode the two most significant bytes, and in the triple-byte mode all the HTX is transferred.

7.4.3 SHI Host Receive Data FIFO (HRX)—DSP Side

The 24-bit Host Receive data FIFO (HRX) is a 10-word deep, First-In-First-Out (FIFO) register used for Host-to-DSP data transfers. The serial data is received via the shift register and then loaded into the HRX. In the single-byte data transfer mode, the most significant byte of the shift register is transferred to the HRX (the other bits are filled with 0s); in the double-byte mode the two most significant bytes are transferred (the least significant byte is filled with 0s), and in the triple-byte mode, all 24 bits are transferred to the HRX. The HRX may be read by the DSP while the FIFO is being loaded from the shift register. Reading all data from HRX will clear the HRNE flag. HRX may be read by DSP core instructions or by DMA transfers. The HRX FIFO is reset to the empty state (cleared) when the chip is in Stop mode, and during hardware reset, software reset, and individual reset.

7.4.4 SHI Slave Address Register (HSAR)—DSP Side

The 24-bit Slave Address Register (HSAR) is used when the SHI operates in the I^2C Slave mode and is ignored in the other operational modes. HSAR holds five bits of the 7-bit slave address of the device. The SHI also acknowledges the general call address (all 0s, 7-bit address, and a 0 R/$\overline{W}$ bit) specified by the I^2C protocol, but will treat any following data bytes as regular data, i.e. the SHI does not differentiate between its dedicated address and the general call address. HSAR cannot be accessed by the host processor.

7.4.4.1 HSAR Reserved Bits—Bits 17–0,19

These bits are reserved and unused. They read as 0s and should be written with 0s for future compatibility.

7.4.4.2 HSAR I^2C Slave Address (HA[6:3], HA1)—Bits 23–20,18

Part of the I^2C slave device address is stored in the read/write HA[6:3], HA1 bits of HSAR. The full 7-bit slave device address is formed by combining the HA[6:3], HA1 bits with the HA0 and HA2 pins to obtain the HA[6:0] slave device address. The full 7-bit slave device address is compared to the received address byte whenever an I^2C master device initiates an I^2C bus transfer. During hardware reset or software reset, HA[6:3] = 1011 while HA1 is cleared; this results in a default slave device address of 1011_HA2_0_HA0.

7.4.5 SHI Clock Control Register (HCKR)—DSP Side

The SHI Clock Control Register (HCKR) is a 24-bit read/write register that controls the SHI clock generator operation. The HCKR bits should be modified only while the SHI is in the individual reset state (HEN = 0 in the HCSR).

NOTE

The programmer should not use the combination HRS = 1 and HDM[7:0] = 00000000, since it may cause synchronization problems and improper operation (it is therefore considered an illegal combination).

NOTE

The HCKR bits are cleared during hardware reset or software reset, except for CPHA, which is set. The HCKR is not affected by the Stop state.

The HCKR bits are described in the following paragraphs.

7.4.5.1 Clock Phase and Polarity (CPHA and CPOL)—Bits 1–0

The programmer may select any of four combinations of Serial Clock (SCK) phase and polarity when operating in the SPI mode (refer to Figure 7-6). The clock polarity is determined by the Clock Polarity (CPOL) control bit, which selects an active-high or active-low clock. When CPOL is cleared, it produces a steady-state low value at the SCK pin of the master device whenever data is not being transferred. If the CPOL bit is set, a high value is produced at the SCK pin of the master device whenever data is not being transferred.

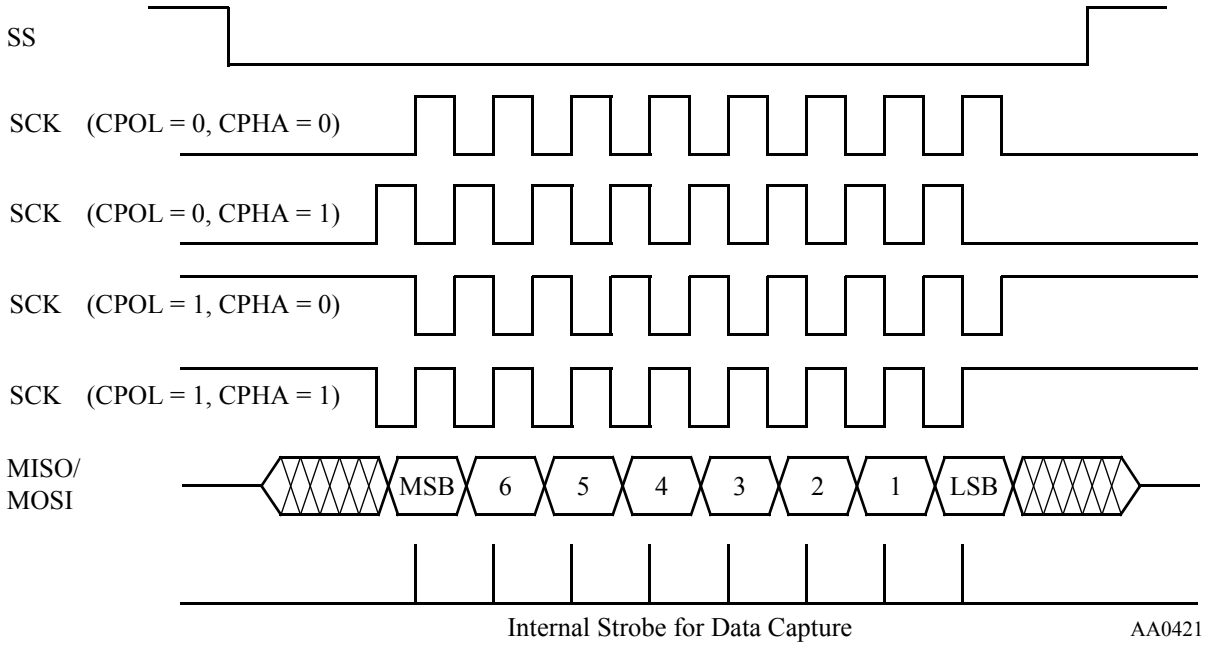


Figure 7-6 SPI Data-To-Clock Timing Diagram

The Clock Phase (CPHA) bit controls the relationship between the data on the MISO and MOSI pins and the clock produced or received at the SCK pin. This control bit is used in conjunction with the CPOL bit to select the desired clock-to-data relationship. The CPHA bit, in general, selects the clock edge that captures data and allows it to change states. It has its greatest impact on the first bit transmitted (MSB) in that it does or does not allow a clock transition before the data capture edge.

When in Slave mode and CPHA = 0, the $\overline{SS}$ line must be deasserted and asserted by the external master between each successive word transfer. $\overline{SS}$ must remain asserted between successive bytes within a word. The DSP core should write the next data word to HTX when HTDE = 1, clearing HTDE. However, the data will be transferred to the shift register for transmission only when $\overline{SS}$ is deasserted. HTDE is set when the data is transferred from HTX to the shift register.

When in Slave mode and CPHA = 1, the $\overline{SS}$ line may remain asserted between successive word transfers. The $\overline{SS}$ must remain asserted between successive bytes within a word. The DSP core should write the next data word to HTX when HTDE = 1, clearing HTDE. The HTX data will be transferred to the shift register for transmission as soon as the shift register is empty. HTDE is set when the data is transferred from HTX to the shift register.

When in Master mode and CPHA = 0, the DSP core should write the next data word to HTX when HTDE = 1, clearing HTDE; the data is transferred immediately to the shift register for transmission. HTDE is set only at the end of the data word transmission.

NOTE

The master is responsible for deasserting and asserting the slave device $\overline{SS}$ line between word transmissions.

When in Master mode and CPHA = 1, the DSP core should write the next data word to HTX when HTDE = 1, clearing HTDE. The HTX data will be transferred to the shift register for transmission as soon as the shift register is empty. HTDE is set when the data is transferred from HTX to the shift register.

The clock phase and polarity should be identical for both the master and slave SPI devices. CPHA and CPOL are functional only when the SHI operates in the SPI mode, and are ignored in the I^2C mode. The CPHA bit is set and the CPOL bit is cleared during hardware reset and software reset.

7.4.5.2 HCKR Prescaler Rate Select (HRS)—Bit 2

The HRS bit controls a prescaler in series with the clock generator divider. This bit is used to extend the range of the divider when slower clock rates are desired. When HRS is set, the prescaler is bypassed. When HRS is cleared, the fixed divide-by-eight prescaler is operational. HRS is ignored when the SHI operates in the Slave mode, except for I^2C when HCKFR is set. The HRS bit is cleared during hardware reset and software reset.

NOTE

Use the equations in the SHI data sheet to determine the value of HRS for the specific serial clock frequency required.

7.4.5.3 HCKR Divider Modulus Select (HDM[7:0])—Bits 10–3

The HDM[7:0] bits specify the divide ratio of the clock generator divider. A divide ratio between 1 and 256 (HDM[7:0] = 0 to $FF) may be selected. When the SHI operates in the Slave mode [except for I^2C when HCKFR is set], the HDM[7:0] bits are ignored. The HDM[7:0] bits are cleared during hardware reset and software reset.

NOTE

Use the equations in the SHI data sheet to determine the value of HDM[7:0] for the specific serial clock frequency required.

7.4.5.4 HCKR Reserved Bits—Bits 23–14, 11

These bits in HCKR are reserved and unused. They are read as 0s and should be written with 0s for future compatibility.

7.4.5.5 HCKR Filter Mode (HFM[1:0]) — Bits 13–12

The read/write control bits HFM[1:0] specify the operational mode of the noise reduction filters, as described in Table 7-3. The filters are designed to eliminate undesired spikes that might occur on the clock and data-in lines and allow the SHI to operate in noisy environments when required. One filter is located in the input path of the SCK/SCL line and the other is located in the input path of the data line (i.e., the SDA line when in I^2C mode, the MISO line when in SPI Master mode, and the MOSI line when in SPI Slave mode).

Table 7-3 SHI Noise Reduction Filter Mode

HFM1	HFM0	Description
0	0	Bypassed (Disabled)
0	1	Reserved
1	0	Narrow Spike Tolerance
1	1	Wide Spike Tolerance

When HFM[1:0] are cleared, the filter is bypassed (spikes are **not** filtered out). This mode is useful when higher bit-rate transfers are required and the SHI operates in a noise-free environment.

When HFM1 = 1 and HFM0 = 0, the narrow-spike-tolerance filter mode is selected. In this mode the filters eliminate spikes with durations of up to 50ns. This mode is suitable for use in mildly noisy environments and imposes some limitations on the maximum achievable bit-rate transfer.

When HFM1 = 1 and HFM0 = 1, the wide-spike-tolerance filter mode is selected. In this mode the filters eliminate spikes up to 100ns. This mode is recommended for use in noisy environments; the bit-rate transfer is strictly limited. The wide-spike- tolerance filter mode is highly recommended for use in I^2C bus systems as it fully conforms to the I^2C bus specification and improves noise immunity.

NOTE

HFM[1:0] are cleared during hardware reset and software reset.

After changing the filter bits in the HCKR to a non-bypass mode (HFM[1:0] not equal to ‘00’), the programmer should wait at least ten times the tolerable spike width before enabling the SHI (setting the HEN bit in the HCSR). Similarly, after changing the I^2C bit in the HCSR or the CPOL bit in the HCKR, while the filter mode bits are in a non-bypass mode (HFM[1:0] not equal to ‘00’), the programmer should wait at least ten times the tolerable spike width before enabling the SHI (setting HEN in the HCSR).

7.4.6 SHI Control/Status Register (HCSR)—DSP Side

The HCSR is a 24-bit read/write register that controls the SHI operation and reflects its status. Each bit is described in one of the following paragraphs. When in the Stop state or during individual reset, the HCSR status bits are reset to their hardware-reset state, while the control bits are not affected.

7.4.6.1 HCSR Host Enable (HEN)—Bit 0

The read/write control bit Host Enable (HEN) enables the overall operation of the SHI. When HEN is set, SHI operation is enabled. When HEN is cleared, the SHI is disabled (individual reset state, see below). The HCKR and the HCSR control bits are not affected when HEN is cleared. When operating in Master mode, HEN should be cleared only after the SHI is idle (HBUSY = 0). HEN is cleared during hardware reset and software reset.

7.4.6.1.1 SHI Individual Reset

While the SHI is in the individual reset state, SHI input pins are inhibited, output and bidirectional pins are disabled (high impedance), the HCSR status bits and the transmit/receive paths are reset to the same state produced by hardware reset or software reset. The individual reset state is entered following a one-instruction-cycle delay after clearing HEN.

7.4.6.2 HCSR I^{2}C/SPI Selection (HI^{2}C)—Bit 1

The read/write control bit HI^{2}C selects whether the SHI operates in the I^{2}C or SPI modes. When HI^{2}C is cleared, the SHI operates in the SPI mode. When HI^{2}C is set, the SHI operates in the I^{2}C mode. HI^{2}C affects the functionality of the SHI pins as described in **Section 2 Pin Descriptions**. It is recommended that an SHI individual reset be generated (HEN cleared) before changing HI^{2}C. HI^{2}C is cleared during hardware reset and software reset.

7.4.6.3 HCSR Serial Host Interface Mode (HM[1:0])—Bits 3–2

The read/write control bits HM[1:0] select the size of the data words to be transferred, as shown in Table 7-4. HM[1:0] should be modified only when the SHI is idle (HBUSY = 0). HM[1:0] are cleared during hardware reset and software reset.

Table 7-4 SHI Data Size

HM1	HMO	Description
0	0	8-bit data
0	1	16-bit data
1	0	24-bit data
1	1	Reserved

7.4.6.4 HCSR I^{2}C Clock Freeze (HCKFR) - Bit 4

The read/write control bit HCSR I^{2}C Clock Freeze (HCKFR) determines the behavior of the SHI when operating in the I^{2}C Slave mode in cases where the SHI is unable to service the master request. The HCKFR bit is used only when operating in the I^{2}C Slave mode; it is ignored otherwise.

If HCKFR is set, the SHI will hold the clock line to GND if it is not ready to send data to the master during a read transfer, or if the input FIFO is full when the master attempts to execute a write transfer. In this way, the master may detect that the slave is not ready for the requested transfer, without causing an error condition in the slave. When HCKFR is set for transmit sessions, the SHI clock generator must be programmed as if to generate the same serial clock as produced by the external master, otherwise erroneous operation may result. The programmed frequency should be equal to or up to 25% less than the external clock frequency.

If HCKFR is cleared, any attempt from the master to execute a transfer when the slave is not ready will result in an overrun or underrun error condition.

It is recommended that an SHI individual reset be generated (HEN cleared) before changing HCKFR. HCKFR is cleared during hardware reset and software reset.

7.4.6.5 HCSR Reserved Bits—Bits 23, 18 and 16

These bits in HCSR are reserved and unused. They are read as 0s and should be written with 0s for future compatibility.

7.4.6.6 HCSR FIFO-Enable Control (HFIFO)—Bit 5

The read/write control bit HCSR FIFO-enable control (HFIFO) selects the size of the receive FIFO. When HFIFO is cleared, the FIFO has a single level. When HFIFO is set, the FIFO has 10 levels. It is recommended that an SHI individual reset be generated (HEN cleared) before changing HFIFO. HFIFO is cleared during hardware reset and software reset.

7.4.6.7 HCSR Master Mode (HMST)—Bit 6

The read/write control bit HCSR Master (HMST) determines the operating mode of the SHI. If HMST is set, the interface operates in the Master mode. If HMST is cleared, the interface operates in the Slave mode. The SHI supports a single-master configuration, in both I^2C and SPI modes. When configured as an SPI Master, the SHI drives the SCK line and controls the direction of the data lines MOSI and MISO. The $\overline{SS}$ line must be held deasserted in the SPI Master mode; if the $\overline{SS}$ line is asserted when the SHI is in SPI Master mode, a bus error will be generated (the HCSR HBER bit will be set—see Section 7.4.6.18, "Host Bus Error (HBER)—Bit 21"). When configured as an I^2C Master, the SHI controls the I^2C bus by generating start events, clock pulses, and stop events for transmission and reception of serial data. It is recommended that an SHI individual reset be generated (HEN cleared) before changing HMST. HMST is cleared during hardware reset and software reset.

7.4.6.8 HCSR Host-Request Enable (HRQE[1:0])—Bits 8–7

The read/write Host-Request Enable control bits (HRQE[1:0]) are used to enable the operation of the $\overline{HREQ}$ pin. When HRQE[1:0] are cleared, the $\overline{HREQ}$ pin is disabled and held in the high impedance state. If either HRQE0 or HRQE1 are set and the SHI is operating in a Master mode, the $\overline{HREQ}$ pin becomes an input that controls SCK: deasserting $\overline{HREQ}$ will suspend SCK. If either HRQE0 or HRQE1 are set and the SHI is operating in a Slave mode, $\overline{HREQ}$ becomes an output and its operation is defined in Table 7-5. HRQE[1:0] should be modified only when the SHI is idle (HBUSY = 0). HRQE[1:0] are cleared during hardware reset and software reset.

Table 7-5 $\overline{HREQ}$ Function In SHI Slave Modes

HRQE1	HRQE0	$\overline{HREQ}$ Pin Operation
0	0	High impedance
0	1	Asserted if IOSR is ready to receive a new word

Table 7-5 $\overline{\text{HREQ}}$ Function In SHI Slave Modes (continued)

HRQE1	HRQE0	$\overline{\text{HREQ}}$ Pin Operation
1	0	Asserted if IOSR is ready to transmit a new word
1	1	I^2C: Asserted if IOSR is ready to transmit or receive SPI: Asserted if IOSR is ready to transmit and receive

7.4.6.9 HCSR Idle (HIDLE)—Bit 9

The read/write control/status bit Host Idle (HIDLE) is used only in the I^2C Master mode; it is ignored otherwise. It is only possible to set the HIDLE bit during writes to the HCSR. HIDLE is cleared by writing to HTX. To ensure correct transmission of the slave device address byte, HIDLE should be set only when HTX is empty (HTDE = 1). After HIDLE is set, a write to HTX will clear HIDLE and cause the generation of a stop event, a start event, and then the transmission of the eight MSBs of the data as the slave device address byte. While HIDLE is cleared, data written to HTX will be transmitted ‘as is.’ If the SHI completes transmitting a word and there is no new data in HTX, the clock will be suspended after sampling ACK. If the SHI completes transmitting a word and there is no new data in HTX when HIDLE is set, a stop event will be generated.

HIDLE determines the acknowledge that the receiver sends after correct reception of a byte. If HIDLE is cleared, the reception will be acknowledged by sending a ‘0’ bit on the SDA line at the ACK clock tick. If HIDLE is set, the reception will not be acknowledged (a ‘1’ bit is sent). It is used to signal an end-of-data to a slave transmitter by not generating an ACK on the last byte. As a result, the slave transmitter must release the SDA line to allow the master to generate the stop event. If the SHI completes receiving a word and the HRX FIFO is full, the clock will be suspended before transmitting an ACK. While HIDLE is cleared the bus is busy, that is, the start event was sent but no Stop event was generated. Setting HIDLE will cause a stop event after receiving the current word.

NOTE

HIDLE is set while the SHI is not in the I^2C Master mode.
Care should be taken to ensure that HIDLE be set by the programmer only after disabling any DMA channel service to HTX.
HIDLE is set during hardware reset, software reset, individual reset, and while the chip is in the Stop state.

7.4.6.10 HCSR Bus-Error Interrupt Enable (HBIE)—Bit 10

The read/write HCSR Bus-error Interrupt Enable (HBIE) control bit is used to enable the SHI bus-error interrupt. If HBIE is cleared, bus-error interrupts are disabled, and the HBER status bit must be polled to determine if an SHI bus error occurred. If both HBIE and HBER are set, the SHI will request SHI bus-error interrupt service from the interrupt controller. HBIE is cleared by hardware reset and software reset.

NOTE

Clearing HBIE will mask a pending bus-error interrupt only after a one-instruction-cycle delay. If HBIE is cleared in a long interrupt service routine, it is recommended that at least one other instruction separate the instruction that clears HBIE and the RTI instruction at the end of the interrupt service routine.

7.4.6.11 HCSR Transmit-Interrupt Enable (HTIE)—Bit 11

The read/write HCSR Transmit-Interrupt Enable (HTIE) control bit is used to enable the SHI transmit data interrupts. If HTIE is cleared, transmit interrupts are disabled, and the HTDE status bit must be polled to determine if the SHI transmit-data register is empty. If both HTIE and HTDE are set and HTUE is cleared, the SHI will request SHI transmit-data interrupt service from the interrupt controller. If both HTIE and HTUE are set, the SHI will request SHI transmit-underrun-error interrupt service from the interrupt controller. HTIE is cleared by hardware reset and software reset.

NOTE

Clearing HTIE will mask a pending transmit interrupt only after a one-instruction cycle-delay. If HTIE is cleared in a long interrupt service routine, it is recommended that at least one other instruction separate the instruction that clears HTIE and the RTI instruction at the end of the interrupt service routine.

7.4.6.12 HCSR Receive Interrupt Enable (HRIE[1:0])—Bits 13–12

The read/write HCSR Receive Interrupt Enable (HRIE[1:0]) control bits are used to enable the SHI receive-data interrupts. If HRIE[1:0] are cleared, receive interrupts are disabled, and the HRNE and HRFF (bits 17 and 19, see below) status bits must be polled to determine if there is data in the receive FIFO. If HRIE[1:0] are not cleared, receive interrupts will be generated according to Table 7-6.

Table 7-6 HCSR Receive Interrupt Enable Bits

HRIE[1:0]	Interrupt	Condition
00	Disabled	Not applicable
01	Receive FIFO not empty Receive Overrun Error	HRNE = 1 and HROE = 0 HROE = 1
10	Reserved	Not applicable
11	Receive FIFO full Receive Overrun Error	HRFF = 1 and HROE = 0 HROE = 1

NOTE

HRIE[1:0] are cleared by hardware and software reset.

NOTE

Clearing HRIE[1:0] will mask a pending receive interrupt only after a one-instruction-cycle delay. If HRIE[1:0] are cleared in a long interrupt service routine, it is recommended that at least one other instruction separate the instruction that clears HRIE[1:0] and the RTI instruction at the end of the interrupt service routine.

7.4.6.13 HCSR Host Transmit Underrun Error (HTUE)—Bit 14

The read-only status bit Host Transmit Underrun Error (HTUE) indicates that a transmit-underrun error occurred. Transmit-underrun errors can occur only when operating in the SPI Slave mode or the I^2C Slave mode when HCKFR is cleared (in a Master mode, transmission takes place on demand and no underrun can occur). It is set when both the shift register and the HTX register are empty and the external master begins reading the next word:

- When operating in the I^2C mode, HTUE is set in the falling edge of the ACK bit. In this case, the SHI will retransmit the previously transmitted word.
- When operating in the SPI mode, HTUE is set at the first clock edge if CPHA = 1; it is set at the assertion of $\overline{SS}$ if CPHA = 0.

If a transmit interrupt occurs with HTUE set, the transmit-underrun interrupt vector will be generated. If a transmit interrupt occurs with HTUE cleared, the regular transmit-data interrupt vector will be generated. HTUE is cleared by reading the HCSR and then writing to the HTX register. HTUE is cleared by hardware reset, software reset, SHI individual reset, and during the Stop state.

7.4.6.14 HCSR Host Transmit Data Empty (HTDE)—Bit 15

The read-only status bit Host Transmit Data Empty (HTDE) indicates that the HTX register is empty and can be written by the DSP. HTDE is set when the data word is transferred from HTX to the shift register, except for a special case in SPI Master mode when CPHA = 0 (see HCKR). When operating in the SPI Master mode with CPHA = 0, HTDE is set after the end of the data word transmission. HTDE is cleared when HTX is written by the DSP either with write instructions or DMA transfers. HTDE is set by hardware reset, software reset, SHI individual reset, and during the Stop state.

7.4.6.15 Host Receive FIFO Not Empty (HRNE)—Bit 17

The read-only status bit Host Receive FIFO Not Empty (HRNE) indicates that the Host Receive FIFO (HRX) contains at least one data word. HRNE is set when the FIFO is not empty. HRNE is cleared when HRX is read by the DSP (read instructions or DMA transfers), reducing the number of words in the FIFO to 0. HRNE is cleared during hardware reset, software reset, SHI individual reset, and during the Stop state.

7.4.6.16 Host Receive FIFO Full (HRFF)—Bit 19

The read-only status bit Host Receive FIFO Full (HRFF) indicates that the Host Receive FIFO (HRX) is full. HRFF is set when the HRX FIFO is full. HRFF is cleared when HRX is read by the DSP (read instructions or DMA transfers) and at least one place is available in the FIFO. HRFF is cleared by hardware reset, software reset, SHI individual reset, and during the Stop state.

7.4.6.17 Host Receive Overrun Error (HROE)—Bit 20

The read-only status bit Host Receive Overrun Error (HROE) indicates that a data-receive overrun error occurred. Receive-overrun errors can not occur when operating in the I^2C Master mode, since the clock is suspended if the receive FIFO is full; it also cannot occur when operating in the I^2C Slave mode when HCKFR is set. HROE is set when the shift register (IOSR) is filled and ready to transfer the data word to the HRX FIFO and the FIFO is already full (HRFF is set). When a receive-overrun error occurs, the shift register is not transferred to the FIFO. If a receive interrupt occurs with HROE set, the receive-overrun interrupt vector will be generated. If a receive interrupt occurs with HROE cleared, the regular receive-data interrupt vector will be generated. HROE is cleared by reading the HCSR with HROE set, followed by reading HRX. HROE is cleared by hardware reset, software reset, SHI individual reset, and during the Stop state.

7.4.6.18 Host Bus Error (HBER)—Bit 21

The read-only status bit Host Bus Error (HBER) indicates that an SHI bus error occurred when operating as a master (HMST set). In I^2C mode, HBER is set if the transmitter does not receive an acknowledge after a byte is transferred; in this case, a stop event will be generated and then transmission will be suspended. In SPI mode, the bit is set if $\overline{SS}$ is asserted; in this case, transmission is suspended at the end of transmission of the current word. HBER is cleared only by hardware reset, software reset, SHI individual reset, and during the Stop state.

7.4.6.19 HCSR Host Busy (HBUSY)—Bit 22

The read-only status bit Host Busy (HBUSY) indicates that the I^2C bus is busy (when in the I^2C mode) or that the SHI itself is busy (when in the SPI mode). When operating in the I^2C mode, HBUSY is set after the SHI detects a Start event and remains set until a Stop event is detected. When operating in the Slave SPI mode, HBUSY is set while $\overline{SS}$ is asserted. When operating in the Master SPI mode, HBUSY is set if the HTX register is not empty or if the IOSR is not empty. HBUSY is cleared otherwise. HBUSY is cleared by hardware reset, software reset, SHI individual reset, and during the Stop state.

7.5 SPI Bus Characteristics

The SPI bus consists of two serial data lines (MISO and MOSI), a clock line (SCK), and a Slave Select line ($\overline{SS}$).During an SPI transfer, a byte is shifted out one data pin while a different byte is simultaneously shifted in through a second data pin. It can be viewed as two 8-bit shift registers connected together in a circular manner, where one shift register is located on the master side and the other on the slave side. Thus the data bytes in the master device and slave device are effectively exchanged. The MISO and MOSI data pins are used for transmitting and receiving serial data. When the SPI is configured as a master, MISO is

the master data input line, and MOSI is the master data output line. When the SPI is configured as a slave device, these pins reverse roles.

Clock control logic allows a selection of clock polarity and a choice of two fundamentally different clocking protocols to accommodate most available synchronous serial peripheral devices. When the SPI is configured as a master, the control bits in the HCKR select the appropriate clock rate, as well as the desired clock polarity and phase format (see Figure 7-6).

The $\overline{SS}$ line allows individual selection of a slave SPI device; slave devices that are not selected do not interfere with SPI bus activity (i.e., they keep their MISO output pin in the high-impedance state). When the SHI is configured as an SPI master device, the $\overline{SS}$ line should be held high. If the $\overline{SS}$ line is driven low when the SHI is in SPI Master mode, a bus error will be generated (the HCSR HBER bit will be set).

7.6 I²C Bus Characteristics

The I²C serial bus consists of two bi-directional lines, one for data signals (SDA) and one for clock signals (SCL). Both the SDA and SCL lines must be connected to a positive supply voltage via a pull-up resistor.

NOTE

Within the I²C bus specifications, the standard mode (100 kHz clock rate) and a fast mode (400 kHz clock rate) are defined. The SHI may operate in both modes.

7.6.1 Overview

The I²C bus protocol must conform to the following rules:

- Data transfer may be initiated only when the bus is not busy.
- During data transfer, the data line must remain stable whenever the clock line is high. Changes in the data line when the clock line is high will be interpreted as control signals (see Figure 7-7).

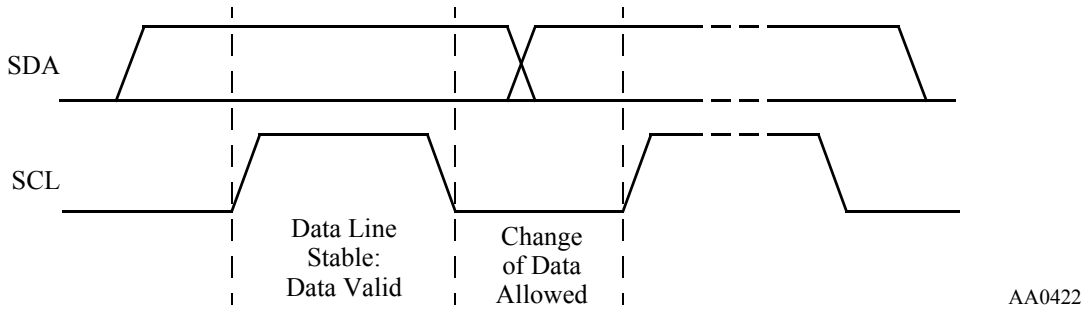


Figure 7-7 I²C Bit Transfer

Accordingly, the I²C bus protocol defines the following events:

- **Bus not busy**—Both data and clock lines remain high.
- **Start data transfer**—The start event is defined as a change in the state of the data line, from high to low, while the clock is high (see Figure 7-8).

- **Stop data transfer**—The stop event is defined as a change in the state of the data line, from low to high, while the clock is high (see Figure 7-8).
- **Data valid**—The state of the data line represents valid data when, after a Start event, the data line is stable for the duration of the high period of the clock signal. The data on the line may be changed during the low period of the clock signal. There is one clock pulse per bit of data.

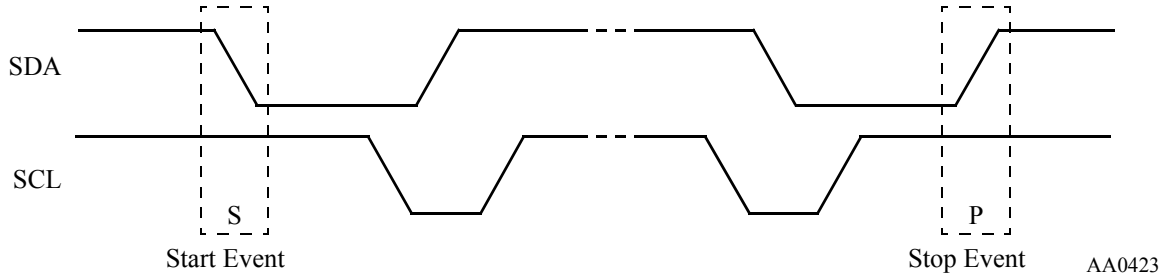


Figure 7-8 I²C Start and Stop Events

Each 8-bit word is followed by one acknowledge bit. This acknowledge bit is a high level put on the bus by the transmitter when the master device generates an extra acknowledge-related clock pulse. A slave receiver that is addressed is obliged to generate an acknowledge after the reception of each byte. Also, a master receiver must generate an acknowledge after the reception of each byte that has been clocked out of the slave transmitter. The device that acknowledges has to pull down the SDA line during the acknowledge clock pulse in such a way that the SDA line is stable low during the high period of the acknowledge-related clock pulse (see Figure 7-9).

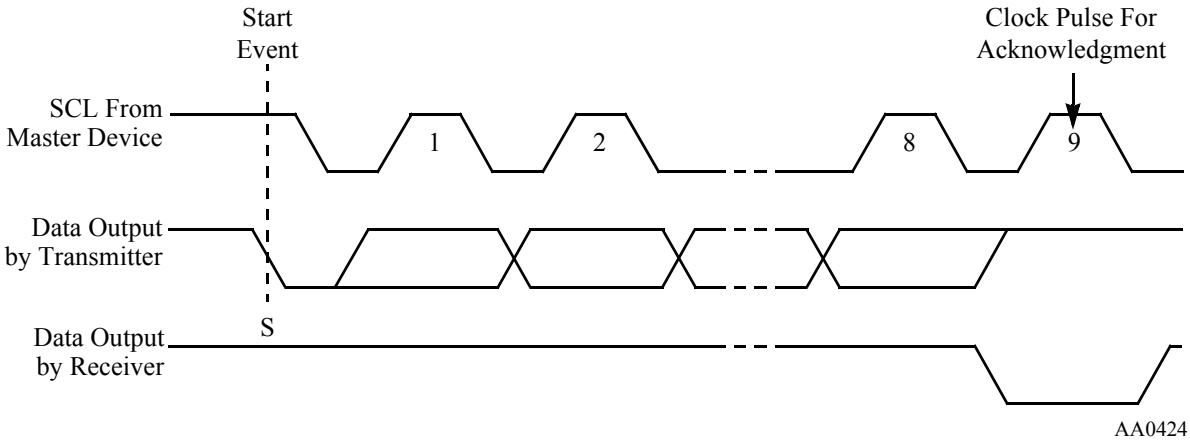


Figure 7-9 Acknowledgment on the I²C Bus

By definition, a device that generates a signal is called a “transmitter,” and the device that receives the signal is called a “receiver.” The device that controls the signal is called the “master” and the devices that are controlled by the master are called “slaves”. A master receiver must signal an end-of-data to the slave transmitter by not generating an acknowledge on the last byte that has been clocked out of the slave device. In this case the transmitter must leave the data line high to enable the master generation of the stop event. Handshaking may also be accomplished by use of the clock synchronizing mechanism. Slave devices can hold the SCL line low, after receiving and acknowledging a byte, to force the master into a wait state until

the slave device is ready for the next byte transfer. The SHI supports this feature when operating as a master device and will wait until the slave device releases the SCL line before proceeding with the data transfer.

7.6.2 I²C Data Transfer Formats

I²C bus data transfers follow the following format: after the start event, a slave device address is sent. This address is 7 bits wide, the eighth bit is a data direction bit (R/$\overline{W}$); '0' indicates a transmission (write), and '1' indicates a request for data (read). A data transfer is always terminated by a stop event generated by the master device. However, if the master device still wishes to communicate on the bus, it can generate another start event, and address another slave device without first generating a stop event (this feature is not supported by the SHI when operating as an I²C master device). This method is also used to provide indivisible data transfers. Various combinations of read/write formats are illustrated in Figure 7-10 and Figure 7-11.

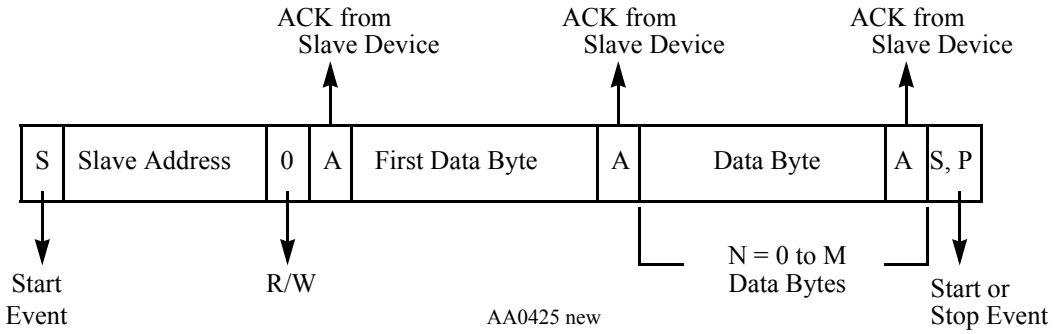


Figure 7-10 I²C Bus Protocol For Host Write Cycle

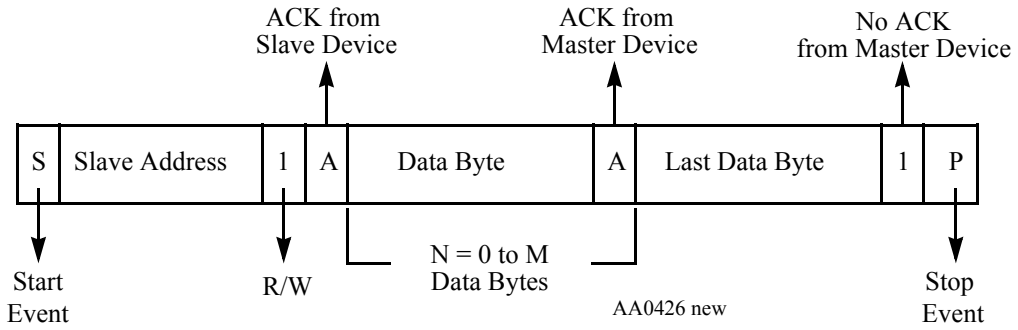


Figure 7-11 I²C Bus Protocol For Host Read Cycle

NOTE

The first data byte in a write-bus cycle can be used as a user-predefined control byte (e.g., to determine the location to which the forthcoming data bytes should be transferred).

7.7 SHI Programming Considerations

The SHI implements both SPI and I^2C bus protocols and can be programmed to operate as a slave device or a single-master device. Once the operating mode is selected, the SHI may communicate with an external device by receiving and/or transmitting data. Before changing the SHI operational mode, an SHI individual reset should be generated by clearing the HEN bit. The following paragraphs describe programming considerations for each operational mode.

7.7.1 SPI Slave Mode

The SPI Slave mode is entered by enabling the SHI (HEN = 1), selecting the SPI mode (HI^2C = 0), and selecting the Slave mode of operation (HMST = 0). The programmer should verify that the CPHA and CPOL bits (in the HCKR) correspond to the external host clock phase and polarity. Other HCKR bits are ignored. When configured in the SPI Slave mode, the SHI external pins operate as follows:

- SCK/SCL is the SCK serial clock input.
- MISO/SDA is the MISO serial data output.
- MOSI/HA0 is the MOSI serial data input.
- $\overline{SS}$/HA2 is the $\overline{SS}$ Slave Select input.
- $\overline{HREQ}$ is the Host Request output.

In the SPI Slave mode, a receive, transmit, or full-duplex data transfer may be performed. Actually, the interface simultaneously performs both data receive and transmit. The status bits of both receive and transmit paths are active; however, the programmer may disable undesired interrupts and ignore non-relevant status bits. It is recommended that an SHI individual reset (HEN cleared) be generated before beginning data reception in order to reset the HRX FIFO to its initial (empty) state (e.g., when switching from transmit to receive data).

If a write to HTX occurs, its contents are transferred to IOSR between data word transfers. The IOSR data is shifted out (via MISO) and received data is shifted in (via MOSI). The DSP may write HTX using either DSP instructions or DMA transfers if the HTDE status bit is set. If no writes to HTX occurred, the contents of HTX are not transferred to IOSR, so the data that is shifted out when receiving is the same as the data present in the IOSR at the time. The HRX FIFO contains valid receive data, which may be read by the DSP using either DSP instructions or DMA transfers, if the HRNE status bit is set.

The $\overline{HREQ}$ output pin, if enabled for receive (HRQE1–HRQE0 = 01), is asserted when the IOSR is ready for receive and the HRX FIFO is not full; this operation guarantees that the next received data word will be stored in the FIFO. The $\overline{HREQ}$ output pin, if enabled for transmit (HRQE1–HRQE0 = 10), is asserted when the IOSR is loaded from HTX with a new data word to transfer. If $\overline{HREQ}$ is enabled for both transmit and receive (HRQE1–HRQE0 = 11), it is asserted when the receive and transmit conditions are true simultaneously. $\overline{HREQ}$ is deasserted at the first clock pulse of the next data word transfer. The $\overline{HREQ}$ line may be used to interrupt the external master device. Connecting the $\overline{HREQ}$ line between two SHI-equipped DSPs, one operating as an SPI master device and the other as an SPI slave device, enables full hardware handshaking if operating with CPHA = 1.

The $\overline{SS}$ line should be kept asserted during a data word transfer. If the $\overline{SS}$ line is deasserted before the end of the data word transfer, the transfer is aborted and the received data word is lost.

7.7.2 SPI Master Mode

The SPI Master mode is initiated by enabling the SHI (HEN = 1), selecting the SPI mode (HI^{2}C = 0), and selecting the Master mode of operation (HMST = 1). Before enabling the SHI as an SPI master device, the programmer should program the proper clock rate, phase, and polarity in HCKR. When configured in the SPI Master mode, the SHI external pins operate as follows:

- SCK/SCL is the SCK serial clock output.
- MISO/SDA is the MISO serial data input.
- MOSI/HA0 is the MOSI serial data output.
- $\overline{\text{SS}}$/HA2 is the SS input. It should be kept deasserted (high) for proper operation.
- $\overline{\text{HREQ}}$ is the Host Request input.

The external slave device can be selected either by using external logic or by activating a GPIO pin connected to its $\overline{\text{SS}}$ pin. However, the $\overline{\text{SS}}$ input pin of the SPI master device should be held deasserted (high) for proper operation. If the SPI master device $\overline{\text{SS}}$ pin is asserted, the Host Bus Error status bit (HBER) is set. If the HBIE bit is also set, the SHI issues a request to the DSP interrupt controller to service the SHI Bus Error interrupt.

In the SPI Master mode the DSP must write to HTX to receive, transmit, or perform a full-duplex data transfer. Actually, the interface performs simultaneous data receive and transmit. The status bits of both receive and transmit paths are active; however, the programmer may disable undesired interrupts and ignore non-relevant status bits. In a data transfer, the HTX is transferred to IOSR, clock pulses are generated, the IOSR data is shifted out (via MOSI) and received data is shifted in (via MISO). The DSP programmer may write HTX (if the HTDE status bit is set) using either DSP instructions or DMA transfers to initiate the transfer of the next word. The HRX FIFO contains valid receive data, which may be read by the DSP using either DSP instructions or DMA transfers, if the HRNE status bit is set.

NOTE

Freescale recommends that an SHI individual reset (HEN cleared) be generated before beginning data reception in order to reset the receive FIFO to its initial (empty) state, such as when switching from transmit to receive data.

The $\overline{\text{HREQ}}$ input pin is ignored by the SPI master device if the HRQE[1:0] bits are cleared, and considered if any of them is set. When asserted by the slave device, $\overline{\text{HREQ}}$ indicates that the external slave device is ready for the next data transfer. As a result, the SPI master sends clock pulses for the full data word transfer. $\overline{\text{HREQ}}$ is deasserted by the external slave device at the first clock pulse of the new data transfer. When deasserted, $\overline{\text{HREQ}}$ will prevent the clock generation of the next data word transfer until it is asserted again. Connecting the $\overline{\text{HREQ}}$ line between two SHI-equipped DSPs, one operating as an SPI master device and the other as an SPI slave device, enables full hardware handshaking if CPHA = 1. For CPHA = 0, $\overline{\text{HREQ}}$ should be disabled by clearing HRQE[1:0].

7.7.3 I^{2}C Slave Mode

The I^{2}C Slave mode is entered by enabling the SHI (HEN = 1), selecting the I^{2}C mode (HI^{2}C = 1), and selecting the Slave mode of operation (HMST = 0). In this operational mode the contents of HCKR are ignored. When configured in the I^{2}C Slave mode, the SHI external pins operate as follows:

- SCK/SCL is the SCL serial clock input.
- MISO/SDA is the SDA open drain serial data line.
- MOSI/HA0 is the HA0 slave device address input.
- $\overline{SS}$/HA2 is the HA2 slave device address input.
- $\overline{HREQ}$ is the Host Request output.

When the SHI is enabled and configured in the I^2C Slave mode, the SHI controller inspects the SDA and SCL lines to detect a start event. Upon detection of the start event, the SHI receives the slave device address byte and enables the slave device address recognition unit. If the slave device address byte was not identified as its personal address, the SHI controller will fail to acknowledge this byte by not driving low the SDA line at the ninth clock pulse (ACK = 1). However, it continues to poll the SDA and SCL lines to detect a new start event. If the personal slave device address was correctly identified, the slave device address byte is acknowledged (ACK = 0 is sent) and a receive/transmit session is initiated according to the eighth bit of the received slave device address byte (i.e., the R/$\overline{W}$ bit).

7.7.3.1 Receive Data in I^2C Slave Mode

A receive session is initiated when the personal slave device address has been correctly identified and the R/$\overline{W}$ bit of the received slave device address byte has been cleared. Following a receive initiation, data in the SDA line is shifted into IOSR MSB first. Following each received byte, an acknowledge (ACK = 0) is sent at the ninth clock pulse via the SDA line. Data is acknowledged bytewise, as required by the I^2C bus protocol, and is transferred to the HRX FIFO when the complete word (according to HM0–HM1) is filled into IOSR. It is the responsibility of the programmer to select the correct number of bytes in an I^2C frame so that they fit in a complete number of words. For this purpose, the slave device address byte does not count as part of the data, and therefore, it is treated separately.

In a receive session, only the receive path is enabled and HTX to IOSR transfers are inhibited. The HRX FIFO contains valid data, which may be read by the DSP using either DSP instructions or DMA transfers if the HRNE status bit is set.

If HCKFR is cleared, when the HRX FIFO is full and IOSR is filled, an overrun error occurs and the HROE status bit is set. In this case, the last received byte will not be acknowledged (ACK = 1 is sent) and the word in the IOSR will not be transferred to the HRX FIFO. This may inform the external I^2C master device of the occurrence of an overrun error on the slave side. Consequently the I^2C master device may terminate this session by generating a stop event.

If HCKFR is set, when the HRX FIFO is full the SHI will hold the clock line to GND not letting the master device write to IOSR. This effectively eliminates the possibility of reaching the overrun condition.

The $\overline{HREQ}$ output pin, if enabled for receive (HRQE1–HRQE0 = 01), is asserted when the IOSR is ready to receive and the HRX FIFO is not full; this operation guarantees that the next received data word will be stored in the FIFO. $\overline{HREQ}$ is deasserted at the first clock pulse of the next received word. The $\overline{HREQ}$ line may be used to interrupt the external I^2C master device. Connecting the $\overline{HREQ}$ line between two SHI-equipped DSPs, one operating as an I^2C master device and the other as an I^2C slave device, enables full hardware handshaking.

7.7.3.2 Transmit Data In I²C Slave Mode

A transmit session is initiated when the personal slave device address has been correctly identified and the R/$\overline{W}$ bit of the received slave device address byte has been set. Following a transmit initiation, the IOSR is loaded from HTX (assuming the latter was not empty) and its contents are shifted out, MSB first, on the SDA line. Following each transmitted byte, the SHI controller samples the SDA line at the ninth clock pulse, and inspects the ACK status. If the transmitted byte was acknowledged (ACK = 0), the SHI controller continues and transmits the next byte. However, if it was not acknowledged (ACK = 1), the transmit session is stopped and the SDA line is released. Consequently, the external master device may generate a stop event in order to terminate the session.

HTX contents are transferred to IOSR when the complete word (according to HM0–HM1) has been shifted out. It is, therefore, the responsibility of the programmer to select the correct number of bytes in an I²C frame so that they fit in a complete number of words. For this purpose, the slave device address byte does not count as part of the data, and therefore, it is treated separately.

In a transmit session, only the transmit path is enabled and the IOSR-to-HRX FIFO transfers are inhibited. When the HTX transfers its valid data word to IOSR, the HTDE status bit is set and the DSP may write a new data word to HTX using either DSP instructions or DMA transfers.

When HCKFR is cleared, if both IOSR and HTX are empty when the master device attempts a transmit session, an underrun condition occurs, setting the HTUE status bit; if this occurs, the previous word will be retransmitted.

When HCKFR is set, if both IOSR and HTX are empty when the master device attempts a transmit session, the SHI will hold the clock line to GND eliminating the possibility of reaching the underrun error condition.

The $\overline{\text{HREQ}}$ output pin, if enabled for transmit (HRQE1–HRQE0 = 10), is asserted when HTX is transferred to IOSR for transmission. When asserted, $\overline{\text{HREQ}}$ indicates that the slave device is ready to transmit the next data word. $\overline{\text{HREQ}}$ is deasserted at the first clock pulse of the next transmitted data word. The $\overline{\text{HREQ}}$ line may be used to interrupt the external I²C master device. Connecting the $\overline{\text{HREQ}}$ line between two SHI-equipped DSPs, one operating as an I²C master device and the other as an I²C slave device, enables full hardware handshaking.

7.7.4 I²C Master Mode

The I²C Master mode is entered by enabling the SHI (HEN = 1), selecting the I²C mode (HI²C = 1) and selecting the Master mode of operation (HMST = 1). Before enabling the SHI as an I²C master, the programmer should program the appropriate clock rate in HCKR.

When configured in the I²C Master mode, the SHI external pins operate as follows:

- SCK/SCL is the SCL serial clock output.
- MISO/SDA is the SDA open drain serial data line.
- MOSI/HA0 is the HA0 slave device address input.
- $\overline{\text{SS}}$/HA2 is the HA2 slave device address input.
- $\overline{\text{HREQ}}$ is the Host Request input.

In the I²C Master mode, a data transfer session is always initiated by the DSP by writing to the HTX register when HIDLE is set. This condition ensures that the data byte written to HTX will be interpreted as being a slave address byte. This data byte must specify the slave device address to be selected and the requested data transfer direction.

NOTE

The slave address byte should be located in the high portion of the data word, whereas the middle and low portions are ignored. Only one byte (the slave address byte) will be shifted out, independent of the word length defined by the HM0–HM1 bits.

In order for the DSP to initiate a data transfer the following actions are to be performed:

- The DSP tests the HIDLE status bit.
- If the HIDLE status bit is set, the DSP writes the slave device address and the R/$\overline{W}$ bit to the most significant byte of HTX.
- The SHI generates a start event.
- The SHI transmits one byte only, internally samples the R/$\overline{W}$ direction bit (last bit), and accordingly initiates a receive or transmit session.
- The SHI inspects the SDA level at the ninth clock pulse to determine the ACK value. If acknowledged (ACK = 0), it starts its receive or transmit session according to the sampled R/$\overline{W}$ value. If not acknowledged (ACK = 1), the HBER status bit in HCSR is set, which will cause an SHI Bus Error interrupt request if HBIE is set, and a stop event will be generated.

The $\overline{HREQ}$ input pin is ignored by the I²C master device if HRQE1 and HRQE0 are cleared, and considered if either of them is set. When asserted, $\overline{HREQ}$ indicates that the external slave device is ready for the next data transfer. As a result, the I²C master device sends clock pulses for the full data word transfer. $\overline{HREQ}$ is deasserted by the external slave device at the first clock pulse of the next data transfer. When deasserted, $\overline{HREQ}$ will prevent the clock generation of the next data word transfer until it is asserted again. Connecting the $\overline{HREQ}$ line between two SHI-equipped DSPs, one operating as an I²C master device and the other as an I²C slave device, enables full hardware handshaking.

7.7.4.1 Receive Data in I²C Master Mode

A receive session is initiated if the R/$\overline{W}$ direction bit of the transmitted slave device address byte is set. Following a receive initiation, data in SDA line is shifted into IOSR MSB first. Following each received byte, an acknowledge (ACK = 0) is sent at the ninth clock pulse via the SDA line if the HIDLE control bit is cleared. Data is acknowledged bytewise, as required by the I²C bus protocol, and is transferred to the HRX FIFO when the complete word (according to HM0–HM1) is filled into IOSR. It is the responsibility of the programmer to select the correct number of bytes in an I²C frame so that they fit in a complete number of words. For this purpose, the slave device address byte does not count as part of the data, and therefore, it is treated separately.

If the I²C slave transmitter is acknowledged, it should transmit the next data byte. In order to terminate the receive session, the programmer should set the HIDLE bit at the last required data word. As a result, the last byte of the next received data word is not acknowledged, the slave transmitter releases the SDA line, and the SHI generates the stop event and terminates the session.

In a receive session, only the receive path is enabled and the HTX-to-IOSR transfers are inhibited. If the HRNE status bit is set, the HRX FIFO contains valid data, which may be read by the DSP using either DSP instructions or DMA transfers. When the HRX FIFO is full, the SHI suspends the serial clock just before acknowledge. In this case, the clock will be reactivated when the FIFO is read (the SHI gives an ACK = 0 and proceeds receiving).

7.7.4.2 Transmit Data In I^2C Master Mode

A transmit session is initiated if the R/$\overline{W}$ direction bit of the transmitted slave device address byte is cleared. Following a transmit initiation, the IOSR is loaded from HTX (assuming HTX is not empty) and its contents are shifted out, MSB-first, on the SDA line. Following each transmitted byte, the SHI controller samples the SDA line at the ninth clock pulse, and inspects the ACK status. If the transmitted byte was acknowledged (ACK = 0), the SHI controller continues transmitting the next byte. However, if it was not acknowledged (ACK = 1), the HBER status bit is set to inform the DSP side that a bus error (or overrun, or any other exception in the slave device) has occurred. Consequently, the I^2C master device generates a stop event and terminates the session.

HTX contents are transferred to the IOSR when the complete word (according to HM0–HM1) has been shifted out. It is, therefore, the responsibility of the programmer to select the right number of bytes in an I^2C frame so that they fit in a complete number of words. Remember that for this purpose, the slave device address byte does not count as part of the data.

In a transmit session, only the transmit path is enabled and the IOSR-to-HRX FIFO transfers are inhibited. When the HTX transfers its valid data word to the IOSR, the HTDE status bit is set and the DSP may write a new data word to HTX using either DSP instructions or DMA transfers. If both IOSR and HTX are empty, the SHI will suspend the serial clock until new data is written into HTX (when the SHI proceeds with the transmit session) or HIDLE is set (the SHI reactivates the clock to generate the Stop event and terminate the transmit session).

7.7.5 SHI Operation During DSP Stop

The SHI operation cannot continue when the DSP is in the Stop state, since no DSP clocks are active. While the DSP is in the stop state, the following will occur:

- If the SHI was operating in the I^2C mode, the SHI pins will be disabled (high impedance).
- If the SHI was operating in the SPI mode, the SHI pins will not be affected.
- The HCSR status bits and the transmit/receive paths are reset to the same state produced by hardware reset or software reset.
- The HCSR and HCKR control bits are not affected.

NOTE

Freescale recommends that the SHI be disabled before entering the Stop state.

NOTES

Appendix A Bootstrap ROM

A.1 DSP56364 Bootstrap Program

```
; BOOTSTRAP CODE FOR DSP56364 - (C) Copyright 1998 Freescale Inc.
; Revised August 11, 1998.
;
;
; This is the Bootstrap program contained in the DSP56364 192-word Boot
; ROM. This program can load any program RAM segment from an external
; EPROM or from the SHI serial interface.
;
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; If MD:MC:MB:MA=0101, then it loads a program RAM segment from consecutive
; byte-wide P memory locations, starting at P:$D00000 (bits 7-0).
; The memory is selected by the Address Attribute AA1 and is accessed with
; 31 wait states.
; The EPROM bootstrap code expects to read 3 bytes
; specifying the number of program words, 3 bytes specifying the address
; to start loading the program words and then 3 bytes for each program
; word to be loaded. The number of words, the starting address and the
; program words are read least significant byte first followed by the
; mid and then by the most significant byte.
; The program words will be condensed into 24-bit words and stored in
; contiguous PRAM memory locations starting at the specified starting address.
; After reading the program words, program execution starts from the same
; address where loading started.
;
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; If MD:MC:MB:MA=0100, then the bootstrap code jumps to the internal
; Program ROM, without loading the Program RAM.
;
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; Operation mode MD:MC:MB:MA=0111 is used for burn-in testing.
;
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; If MD:MC:MB:MA=11xx, then the Program RAM is loaded from the SHI.
;
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;

        page    132,55,0,0,0
```

```
        opt     cex,mex,mu

;;
;;;;;;;;;;;;;;;;;;; GENERAL EQUATES ;;;;;;;;;;;;;;;;;;;;;;;;
;;
BOOT    equ     $D00000         ; this is the location in P memory
                                ; on the external memory bus
                                ; where the external byte-wide
                                ; EPROM would be located
AARV    equ     $D00409         ; AAR1 selects the EPROM as CE~
                                ; mapped as P from $D00000 to
                                ; $DFFFFF, active low
PROMADDR equ    $FF1000         ; Starting PROM address

MA      EQU     0
MB      EQU     1
MC      EQU     2
MD      EQU     3

;;
;;;;;;;;;;;;;;;;;;; DSP I/O REGISTERS ;;;;;;;;;;;;;;;;;;;;;;;;
;;

M_AAR1  EQU     $FFFFF8         ; Address Attribute Register 1

M_HRX   EQU     $FFFF94         ; SHI Receive FIFO
M_HCSR  EQU     $FFFF91         ; SHI Control/Status Register
hrne    equ     17              ; SHI FIFO Not Empty flag
hi2c    equ     1               ; SHI I2C Enable Control Bit
hckfr   equ     4               ; SHI I2C Clock Freeze Control Bit

        ORG PL:$ff0000,PL:$ff0000    ; bootstrap code starts at $ff0000

START
        clr a #$0,r5            ; clear a and init R5 with 0
        jclr #MC,omr,OMRX0XX    ; If MD:MC:MB:MA=x0xx, go to OMRX0XX
        jset #MD,omr,SHILD      ; If MD:MC:MB:MA=11xx, go load from SHI
        jset #MB,omr,BURN_RES   ; If MD:MC:MB:MA=011X, go to BURN/RESERV
        jset #MA,omr,EPROMLD    ; If MD:MC:MB:MA=0101, go load from EPROM

;=====================================================================
; This is the routine that jumps to the internal Program ROM.
; MD:MC:MB:MA=0100

        move #PROMADDR,r1       ; store starting PROM address in r1

        bra     <FINISH

;=====================================================================
; This is the routine that loads from SHI.
; MD:MC:MB:MA=1100 - reserved
; MD:MC:MB:MA=1101 - Bootstrap from SHI (SPI slave)
```

```
; MD:MC:MB:MA=1110 - Bootstrap from SHI (I2C slave, HCKFR=1)
; MD:MC:MB:MA=1111 - Bootstrap from SHI (I2C slave, HCKFR=0)

SHILD

; This is the routine which loads a program through the SHI port.
; The SHI operates in the slave
; mode, with the 10-word FIFO enabled, and with the HREQ pin enabled for
; receive operation. The word size for transfer is 24 bits. The SHI
; operates in the SPI or in the I2C mode, according to the bootstrap mode.
;
; The program is downloaded according to the following rules:
; 1) 3 bytes - Define the program length.
; 2) 3 bytes - Define the address to which to start loading the program to.
; 3) 3n bytes (while n is the program length defined by the first 3 bytes)
; The program words will be stored in contiguous PRAM memory locations starting
; at the specified starting address.
; After storing the program words, program execution starts from the same
; address where loading started.

               move   #$A9,r1          ; prepare SHI control value in r1

; HEN=1, HI2C=0, HM1-HM0=10, HCKFR=0, HFIFO=1, HMST=0,
; HRQE1-HRQE0=01, HIDLE=0, HBIE=0, HTIE=0, HRIE1-HRIE0=00

              jclr    #MA,omr,SHI_CF   ; If MD:MC:MB:MA=11x0, go to SHI clock freeze

               jclr    #MB,omr,shi_loop   ; If MD:MC:MB:MA=1101, select SPI mode

               bset    #hi2c,r1           ; otherwise select I2C mode.

shi_loop       movep   r1,x:M_HCSR        ; enable SHI

               jclr    #hrne,x:M_HCSR,*   ; wait for no. of words
               movep   x:M_HRX,a0

               jclr    #hrne,x:M_HCSR,*   ; wait for starting address
               movep   x:M_HRX,r0
               move    r0,r1

               do      a0,_LOOP2
               jclr    #hrne,x:M_HCSR,*   ; wait for HRX not empty
               movep   x:M_HRX,p:(r0)+    ; store in Program RAM
               nop                        ; req. because of restriction
_LOOP2

        bra     <FINISH

SHI_CF
              bset   #hi2c,r1             ; select I2C mode.
              bset    #hckfr,r1           ; enable clock freeze in I2C mode.
 jset    #MB,omr,shi_loop        ; If MD:MC:MB:MA=1110, go to I2C load
 bra     <RESERVED          ; If MD:MC:MB:MA=1100, go to reserved
```

```
;=====================================================================
; This is the routine that loads from external EPROM.
; MD:MC:MB:MA=0101

EPROMLD

        move #BOOT,r2           ; r2 = address of external EPROM
        movep #AARV,X:M_AAR1    ; aar1 configured for SRAM types of access
        do #6,_LOOP9            ; read number of words and starting address
        movem p:(r2)+,a2        ; Get the 8 LSB from ext. P mem.
        asr #8,a,a              ; Shift 8 bit data into A1
_LOOP9                          ;
        move a1,r0              ; starting address for load
        move a1,r1              ; save it in r1
                                ; a0 holds the number of words
        do a0,_LOOP10           ; read program words
        do #3,_LOOP11           ; Each instruction has 3 bytes
        movem p:(r2)+,a2        ; Get the 8 LSB from ext. P mem.
        asr #8,a,a              ; Shift 8 bit data into A1
_LOOP11                         ; Go get another byte.
        movem a1,p:(r0)+        ; Store 24-bit result in P mem.
        nop                     ; pipeline delay
_LOOP10                         ; and go get another 24-bit word.
                                ; Boot from EPROM done

;=====================================================================
; This is the exit handler that returns execution to normal
; expanded mode and jumps to the RESET vector.

FINISH
        andi #$0,ccr            ; Clear CCR as if RESET to 0.
        jmp (r1)                ; Then go to starting Prog addr.


;=====================================================================
; MD:MC:MB:MA=0111 is Burn-in code
; MD:MC:MB:MA=0110 is reserved


BURN_RES
        jset #MA,omr,BURN       ; If MD:MC:MB:MA=0111, go to BURN


;=====================================================================
; The following modes are reserved:
; MD:MC:MB:MA=0110 is reserved
; MD:MC:MB:MA=1100 is reserved
; MD:MC:MB:MA=X0XX are reserved

OMRX0XX
RESERVED
```

```
       bra     <*

;=========================================================================
; Code for burn-in
;=========================================================================

M_OGDB  EQU     $FFFFFC         ;; OnCE GDB Register
M_PCRC  EQU     $FFFFBF         ;; Port C GPIO Control Register
M_PDRC  EQU     $FFFFBD         ;; Port C GPIO Data Register
M_PRRC  EQU     $FFFFBE         ;; Port C Direction Register
SCKT    EQU     $3              ;; SCKT is GPIO bit #3 in ESAI (Port C)


EQUALDATA       equ     0       ;; 1 if xram and yram are of equal
                                ;; size and addresses, 0 otherwise.

        if      (EQUALDATA)
start_dram      equ     0       ;;
length_dram     equ     $1600   ;; same addresses
        else
start_xram      equ     0       ;; 1k XRAM
length_xram     equ     $0400
start_yram      equ     0       ;; 1.5k YRAM
length_yram     equ     $0600
        endif

start_pram      equ     0       ;; 0.5k PRAM
length_pram     equ     $0200


BURN
                ;; get PATTERN pointer
                clr b   #PATTERNS,r6            ;; b is the error accumulator
                move    #<(NUM_PATTERNS-1),m6   ;; program runs forever in
                                                ;; cyclic form

                ;; configure SCKT as gpio output.
                movep   b,x:M_PDRC              ;; clear GPIO data register
                   bclr    #SCKT,x:M_PCRC           ;; Define SCKT as output GPIO pin
                bset    #SCKT,x:M_PRRC           ;; SCKT toggles means test pass


                                                ;; r5 = test fail flag = $000000
                lua     (r5)-,r7                ;; r7 = test pass flag = $FFFFFF
burnin_loop
do #9,burn1
                ;;---------------------------
                ;; test RAM
                ;;  each pass checks 1 pattern
                ;;---------------------------
                move    p:(r6)+,x1              ;; pattern for x memory
                move    p:(r6)+,x0              ;; pattern for y memory
                move    p:(r6)+,y0              ;; pattern for p memory
```

```
                ;; write pattern to all memory locations

        if      (EQUALDATA)                     ;; x/y ram symmetrical
                ;; write x and y memory
                clr a   #start_dram,r0          ;; start of x/y ram
                move    #>length_dram,n0        ;; length of x/y ram
                rep     n0
                mac x0,x1,a x,l:(r0)+           ;; exercise mac, write x/y ram

        else                                    ;; x/y ram not symmetrical

                ;; write x memory
                clr a   #start_xram,r0          ;; start of xram
                move    #>length_xram,n0        ;; length of xram
                rep     n0
                mac x0,y0,a x1,x:(r0)+          ;; exercise mac, write xram

                ;; write y memory
                clr a   #start_yram,r1          ;; start of yram
                move    #>length_yram,n1        ;; length of yram
                rep     n1
                mac x1,y0,a x0,y:(r1)+          ;; exercise mac, write yram

        endif

                ;; write p memory
                clr a   #start_pram,r2          ;; start of pram
                move    #>length_pram,n2        ;; length of pram
                rep     n2
                move    y0,p:(r2)+              ;; write pram

                ;; check memory contents

        if      (EQUALDATA)                     ;; x/y ram symmetrical

                ;; check dram
                clr a   #start_dram,r0          ;; restore pointer, clear a
                do      n0,_loopd
                move    x:(r0),a1               ;; a0=a2=0
                eor     x1,a
                add     a,b                     ;; accumulate error in b
                move    y:(r0)+,a1              ;; a0=a2=0
                eor     x0,a
                add     a,b                     ;; accumulate error in b
_loopd

        else                                    ;; x/y ram not symmetrical

                ;; check xram
                clr a   #start_xram,r0          ;; restore pointer, clear a
                do      n0,_loopx
                move    x:(r0)+,a1              ;; a0=a2=0
```

```
                eor     x1,a
                add     a,b                     ;; accumulate error in b
_loopx

                ;; check yram
                clr a   #start_yram,r1          ;; restore pointer, clear a
                do      n1,_loopy
                move    y:(r1)+,a1              ;; a0=a2=0
                eor     x0,a
                add     a,b                     ;; accumulate error in b
_loopy

        endif

                ;; check pram
                clr a   #start_pram,r2          ;; restore pointer, clear a
                do      n2,_loopp
                move    p:(r2)+,a1              ;; a0=a2=0
                eor     y0,a
                add     a,b                     ;; accumulate error in b
_loopp
                ;;-------------------------------------------------
                ;; toggle pin if no errors, stop execution otherwise.
                ;;-------------------------------------------------
                ;; if error
                tne     r5,r7                   ;; r7=$FFFFFF as long as test pass
                                                ;; condition codes preserved
                                        ;; this instr can be removed in case of shortage
                movep   r7,x:M_OGDB             ;; write pass/fail flag to OnCE
                                                ;; condition codes preserved
                                        ;; this instr can be removed in case of shortage
                beq     label1
                bclr    #SCKT,x:M_PDRC          ;; clear SCKT if error,
                enddo                           ;; terminate the loop normally
                                        ;; this instr can be removed in case of shortage
                bra     <burn1                  ;; and stop execution
label1                                          ;; if no error
                bchg    #SCKT,x:M_PDRC          ;; toggle pin and keep on looping

burn1                                           ;; test completion
              debug                         ;; enter debug mode if OnCE port enabled
                                        ;; this instr can be removed in case of shortage
             wait                          ;; enter wait otherwise (OnCE port disabled)

BURN_END


                ORG PL:,PL:
PATTERNS        dsm     4                   ;; align for correct modulo addressing

               ORG  PL:BURN_END,PL:BURN_END

            dup PATTERNS-*                  ; write address in unused Boot ROM location
```

```
                dc *
                endm


               ORG     PL:PATTERNS,PL:PATTERNS ;; Each value is written to all memories

                 dc      $555555
                 dc      $AAAAAA
                 dc      $333333
                 dc      $F0F0F0

NUM_PATTERNS     equ     *-PATTERNS


;=====================================================================
; This code fills the unused bootstrap rom locations with their address

                dup $FF00C0-*
                dc *
                endm



;=====================================================================
; Reserved Area in the Program ROM: upper 128 words.
; Address range: $FF2F80 - $FF2FFF
;=====================================================================


                ORG PL:$FF2F80,PL:$FF2F80

; This code fills the unused rom locations with their address

                dup $FF3000-*
                dc *
                endm


                end
```

Appendix B BDSL File

B.1 BSDL FILE

```
-- M O T O R O L A   S S D T   J T A G   S O F T W A R E
-- BSDL File Generated: Sun Aug 16 10:48:15 1998
--
-- Revision History:
--

entity DSP56364 is
generic (PHYSICAL_PIN_MAP : string := "TQFP100");

port (  TCK: in bit;
        TMS: in bit;
        TDI: in bit;
        TDO: out bit;
       MODD: in bit;
       MODB: in bit;
       MODA: in bit;
        FST: inout bit;
        FSR: inout bit;
       SCKT: inout bit;
       SCKR: inout bit;
       SVCC: linkage bit_vector(0 to 2);
       SGND: linkage bit_vector(0 to 2);
       HCKT: inout bit;
      QVCCL: linkage bit_vector(0 to 3);
       QGND: linkage bit_vector(0 to 3);
       HCKR: inout bit;
       SDO0: inout bit;
      QVCCH: linkage bit_vector(0 to 3);
       SDO1: inout bit;
     SDOI23: inout bit;
     SD0I32: inout bit;
     SDOI41: inout bit;
     SDOI50: inout bit;
       SS_N: in bit;
       MOSI: inout bit;
        SDA: inout bit;
        SCK: inout bit;
     HREQ_N: inout bit;
      NMI_N: in bit;
    RESET_N: in bit;
       PVCC: linkage bit;
       PCAP: linkage bit;
       PGND: linkage bit;
```

```
      EXTAL: in bit;
       TA_N: in bit;
      CAS_N: out bit;
       WR_N: out bit;
       RD_N: out bit;
       CVCC: linkage bit;
       CGND: linkage bit;
        AA1: out bit;
        AA0: out bit;
          A: out bit_vector(0 to 17);
       AGND: linkage bit_vector(0 to 3);
       AVCC: linkage bit_vector(0 to 3);
       DVCC: linkage bit;
       DGND: linkage bit;
       GPIO: inout bit_vector(0 to 3);
          R: linkage bit_vector(1 to 4);
          D: inout bit_vector(0 to 7));

use STD_1149_1_1994.all;

attribute COMPONENT_CONFORMANCE of DSP56364 : entity is "STD_1149_1_1993";

attribute PIN_MAP of DSP56364 : entity is PHYSICAL_PIN_MAP;

constant TQFP100 : PIN_MAP_STRING :=
         "MODD:      1, " &
         "MODB:      2, " &
         "MODA:      3, " &
         "FST:       4, " &
         "FSR:       5, " &
         "SCKT:      6, " &
         "SCKR:      7, " &
         "SVCC:     (8, 21, 92), " &
         "SGND:     (9, 22, 93), " &
         "HCKT:      10, " &
         "QVCCL:    (11, 37, 66, 87), " &
         "QGND:     (12, 36, 65, 88), " &
         "HCKR:      13, " &
         "SDO0:      14, " &
         "QVCCH:    (15, 35, 61, 89), " &
         "SDO1:      16, " &
         "SDOI23:    17, " &
         "SD0I32:    18, " &
         "SDOI41:    19, " &
         "SDOI50:    20, " &
         "SS_N:      23, " &
         "MOSI:      24, " &
         "SDA:       25, " &
         "SCK:       26, " &
         "HREQ_N:    27, " &
         "NMI_N:     28, " &
         "RESET_N:   29, " &
         "R:        (30, 85, 86, 90), " &
```

```
          "PVCC:       31, " &
          "PCAP:       32, " &
          "PGND:       33, " &
          "EXTAL:      34, " &
          "TA_N:       38, " &
          "CAS_N:      39, " &
          "WR_N:       40, " &
          "RD_N:       41, " &
          "CVCC:       42, " &
          "CGND:       43, " &
          "AA1:        44, " &
          "AA0:        45, " &
          "A:         (46, 47, 50, 51, 52, 53, 54, 57, 58, 59, 60, 62, 67, 68, 69, 70,
73, 74), " &
          "AVCC:      (48, 55, 63, 71), " &
          "AGND:      (49, 56, 64, 72), " &
          "D:         (75, 76, 77, 78, 81, 82, 83, 84), " &
          "DVCC:       79, " &
          "DGND:       80, " &
          "GPIO:      (91, 94, 95, 96), " &
          "TDO:        97, " &
          "TDI:        98, " &
          "TCK:        99, " &
          "TMS:        100 ";

attribute TAP_SCAN_IN    of  TDI : signal is true;
attribute TAP_SCAN_OUT   of  TDO : signal is true;
attribute TAP_SCAN_MODE  of  TMS : signal is true;
attribute TAP_SCAN_CLOCK of  TCK : signal is (20.0e6, BOTH);

attribute INSTRUCTION_LENGTH of DSP56364 : entity is 4;

attribute INSTRUCTION_OPCODE of DSP56364 : entity is
   "EXTEST       (0000)," &
   "SAMPLE       (0001)," &
   "IDCODE       (0010)," &
   "CLAMP        (0011)," &
   "HIGHZ        (0100)," &
   "ONCE_ENABLE  (0110)," &
   "DEBUG_REQUEST(0111)," &
   "BYPASS       (1111)";

attribute INSTRUCTION_CAPTURE of DSP56364 : entity is "0001";
attribute IDCODE_REGISTER   of DSP56364 : entity is
   "0000"          & -- version
   "000110"        & -- manufacturer's use
   "0001100100"    & -- sequence number
   "00000001110"   & -- manufacturer identity
   "1";              -- 1149.1 requirement


attribute REGISTER_ACCESS of DSP56364 : entity is
   "ONCE[8]    (ONCE_ENABLE,DEBUG_REQUEST)" ;
```

```
attribute BOUNDARY_LENGTH of DSP56364 : entity is 86;

attribute BOUNDARY_REGISTER of DSP56364 : entity is
-- num    cell   port    func           safe [ccell  dis  rslt]
   "0     (BC_1, *,         control,       1)," &
   "1     (BC_6, GPIO(3),   bidir,         X,      0,   1,   Z)," &
   "2     (BC_1, *,         control,       1)," &
   "3     (BC_6, GPIO(2),   bidir,         X,      2,   1,   Z)," &
   "4     (BC_1, *,         control,       1)," &
   "5     (BC_6, GPIO(1),   bidir,         X,      4,   1,   Z)," &
   "6     (BC_1, *,         control,       1)," &
   "7     (BC_6, GPIO(0),   bidir,         X,      6,   1,   Z)," &
   "8     (BC_6, D(7),      bidir,         X,     12,   1,   Z)," &
   "9     (BC_6, D(6),      bidir,         X,     12,   1,   Z)," &
   "10    (BC_6, D(5),      bidir,         X,     12,   1,   Z)," &
   "11    (BC_6, D(4),      bidir,         X,     12,   1,   Z)," &
   "12    (BC_1, *,         control,       1)," &
   "13    (BC_6, D(3),      bidir,         X,     12,   1,   Z)," &
   "14    (BC_6, D(2),      bidir,         X,     12,   1,   Z)," &
   "15    (BC_6, D(1),      bidir,         X,     12,   1,   Z)," &
   "16    (BC_6, D(0),      bidir,         X,     12,   1,   Z)," &
   "17    (BC_1, A(17),     output3,       X,     23,   1,   Z)," &
   "18    (BC_1, A(16),     output3,       X,     23,   1,   Z)," &
   "19    (BC_1, A(15),     output3,       X,     23,   1,   Z)," &
-- num    cell   port    func           safe [ccell  dis  rslt]
   "20    (BC_1, A(14),     output3,       X,     23,   1,   Z)," &
   "21    (BC_1, A(13),     output3,       X,     23,   1,   Z)," &
   "22    (BC_1, A(12),     output3,       X,     23,   1,   Z)," &
   "23    (BC_1, *,         control,       1)," &
   "24    (BC_1, A(11),     output3,       X,     23,   1,   Z)," &
   "25    (BC_1, A(10),     output3,       X,     23,   1,   Z)," &
   "26    (BC_1, A(9),      output3,       X,     23,   1,   Z)," &
   "27    (BC_1, A(8),      output3,       X,     29,   1,   Z)," &
   "28    (BC_1, A(7),      output3,       X,     29,   1,   Z)," &
   "29    (BC_1, *,         control,       1)," &
   "30    (BC_1, A(6),      output3,       X,     29,   1,   Z)," &
   "31    (BC_1, A(5),      output3,       X,     29,   1,   Z)," &
   "32    (BC_1, A(4),      output3,       X,     29,   1,   Z)," &
   "33    (BC_1, A(3),      output3,       X,     29,   1,   Z)," &
   "34    (BC_1, A(2),      output3,       X,     29,   1,   Z)," &
   "35    (BC_1, A(1),      output3,       X,     29,   1,   Z)," &
   "36    (BC_1, A(0),      output3,       X,     29,   1,   Z)," &
   "37    (BC_1, *,         control,       1)," &
   "38    (BC_1, AA0,       output3,       X,     37,   1,   Z)," &
   "39    (BC_1, *,         control,       1)," &
-- num    cell   port    func           safe [ccell  dis  rslt]
   "40    (BC_1, AA1,       output3,       X,     39,   1,   Z)," &
   "41    (BC_1, *,         control,       1)," &
   "42    (BC_1, RD_N,      output3,       X,     41,   1,   Z)," &
   "43    (BC_1, WR_N,      output3,       X,     41,   1,   Z)," &
   "44    (BC_1, *,         control,       1)," &
   "45    (BC_1, CAS_N,     output3,       X,     44,   1,   Z)," &
```

```
   "46    (BC_1, TA_N,     input,         X)," &
   "47    (BC_1, EXTAL,    input,         X)," &
   "48    (BC_1, RESET_N,  input,         X)," &
   "49    (BC_1, NMI_N,    input,         X)," &
   "50    (BC_1, *,        control,       1)," &
   "51    (BC_6, HREQ_N,   bidir,         X,      50,   1,   Z)," &
   "52    (BC_1, *,        control,       1)," &
   "53    (BC_6, SCK,      bidir,         X,      52,   1,   Z)," &
   "54    (BC_1, *,        control,       1)," &
   "55    (BC_6, SDA,      bidir,         X,      54,   1,   Z)," &
   "56    (BC_1, *,        control,       1)," &
   "57    (BC_6, MOSI,     bidir,         X,      56,   1,   Z)," &
   "58    (BC_1, SS_N,     input,         X)," &
   "59    (BC_1, *,        control,       1)," &
-- num    cell   port   func           safe [ccell  dis  rslt]
   "60    (BC_6, SDOI50,   bidir,         X,      59,   1,   Z)," &
   "61    (BC_1, *,        control,       1)," &
   "62    (BC_6, SDOI41,   bidir,         X,      61,   1,   Z)," &
   "63    (BC_1, *,        control,       1)," &
   "64    (BC_6, SD0I32,   bidir,         X,      63,   1,   Z)," &
   "65    (BC_1, *,        control,       1)," &
   "66    (BC_6, SDOI23,   bidir,         X,      65,   1,   Z)," &
   "67    (BC_1, *,        control,       1)," &
   "68    (BC_6, SDO1,     bidir,         X,      67,   1,   Z)," &
   "69    (BC_1, *,        control,       1)," &
   "70    (BC_6, SDO0,     bidir,         X,      69,   1,   Z)," &
   "71    (BC_1, *,        control,       1)," &
   "72    (BC_6, HCKR,     bidir,         X,      71,   1,   Z)," &
   "73    (BC_1, *,        control,       1)," &
   "74    (BC_6, HCKT,     bidir,         X,      73,   1,   Z)," &
   "75    (BC_1, *,        control,       1)," &
   "76    (BC_6, SCKR,     bidir,         X,      75,   1,   Z)," &
   "77    (BC_1, *,        control,       1)," &
   "78    (BC_6, SCKT,     bidir,         X,      77,   1,   Z)," &
   "79    (BC_1, *,        control,       1)," &
-- num    cell   port   func           safe [ccell  dis  rslt]
   "80    (BC_6, FSR,      bidir,         X,      79,   1,   Z)," &
   "81    (BC_1, *,        control,       1)," &
   "82    (BC_6, FST,      bidir,         X,      81,   1,   Z)," &
   "83    (BC_1, MODA,     input,         X)," &
   "84    (BC_1, MODB,     input,         X)," &
   "85    (BC_1, MODD,     input,         X)";

end DSP56364;
```

Appendix C Programmer's Reference

C.1 Introduction

This section has been compiled as a reference for programmers. It contains a table showing the addresses of all the DSPs memory-mapped peripherals, an interrupt address table, an interrupt exception priority table, a quick reference to the host interface, and programming sheets for the major programmable registers on the DSP.

C.1.1 Peripheral Addresses

Table C-1 lists the memory addresses of all on-chip peripherals.

C.1.2 Interrupt Addresses

Table C-2 lists the interrupt starting addresses and sources.

C.1.3 Interrupt Priorities

Table C-3 lists the priorities of specific interrupts within interrupt priority levels.

Table C-1. Internal I/O Memory Map

Peripheral	Address	Register Name
IPR	$FFFFFF	INTERRUPT PRIORITY REGISTER CORE (IPR-C)
	$FFFFFE	INTERRUPT PRIORITY REGISTER PERIPHERAL (IPR-P)
PLL	$FFFFFD	PLL CONTROL REGISTER (PCTL)
ONCE	$FFFFFC	ONCE GDB REGISTER (OGDB)
BIU	$FFFFFB	BUS CONTROL REGISTER (BCR)
	$FFFFFA	DRAM CONTROL REGISTER (DCR)
	$FFFFF9	ADDRESS ATTRIBUTE REGISTER 0 (AAR0)
	$FFFFF8	ADDRESS ATTRIBUTE REGISTER 1 (AAR1)
	$FFFFF7	ADDRESS ATTRIBUTE REGISTER 2 (AAR2)
	$FFFFF6	ADDRESS ATTRIBUTE REGISTER 3 (AAR3)
	$FFFFF5	ID REGISTER (IDR)

Table C-1. Internal I/O Memory Map (continued)

Peripheral	Address	Register Name
DMA	$FFFFF4	DMA STATUS REGISTER (DSTR)
	$FFFFF3	DMA OFFSET REGISTER 0 (DOR0)
	$FFFFF2	DMA OFFSET REGISTER 1 (DOR1)
	$FFFFF1	DMA OFFSET REGISTER 2 (DOR2)
	$FFFFF0	DMA OFFSET REGISTER 3 (DOR3)
DMA0	$FFFFEF	DMA SOURCE ADDRESS REGISTER (DSR0)
	$FFFFEE	DMA DESTINATION ADDRESS REGISTER (DDR0)
	$FFFFED	DMA COUNTER (DCO0)
	$FFFFEC	DMA CONTROL REGISTER (DCR0)
DMA1	$FFFFEB	DMA SOURCE ADDRESS REGISTER (DSR1)
	$FFFFEA	DMA DESTINATION ADDRESS REGISTER (DDR1)
	$FFFFE9	DMA COUNTER (DCO1)
	$FFFFE8	DMA CONTROL REGISTER (DCR1)
DMA2	$FFFFE7	DMA SOURCE ADDRESS REGISTER (DSR2)
	$FFFFE6	DMA DESTINATION ADDRESS REGISTER (DDR2)
	$FFFFE5	DMA COUNTER (DCO2)
	$FFFFE4	DMA CONTROL REGISTER (DCR2)
DMA3	$FFFFE3	DMA SOURCE ADDRESS REGISTER (DSR3)
	$FFFFE2	DMA DESTINATION ADDRESS REGISTER (DDR3)
	$FFFFE1	DMA COUNTER (DCO3)
	$FFFFE0	DMA CONTROL REGISTER (DCR3)
DMA4	$FFFFDF	DMA SOURCE ADDRESS REGISTER (DSR4)
	$FFFFDE	DMA DESTINATION ADDRESS REGISTER (DDR4)
	$FFFFDD	DMA COUNTER (DCO4)
	$FFFFDC	DMA CONTROL REGISTER (DCR4)
DMA5	$FFFFDB	DMA SOURCE ADDRESS REGISTER (DSR5)
	$FFFFDA	DMA DESTINATION ADDRESS REGISTER (DDR5)
	$FFFFD9	DMA COUNTER (DCO5)
	$FFFFD8	DMA CONTROL REGISTER (DCR5)

Table C-1. Internal I/O Memory Map (continued)

Peripheral	Address	Register Name
Reserved	$FFFFD7 thru $FFFFD0	RESERVED
PORT B	$FFFFCF	PORT B CONTROL REGISTER (PCRB)
	$FFFFCE	PORT B DIRECTION REGISTER (PRRB)
	$FFFFCD	PORT B GPIO DATA REGISTER (PDRB)
Reserved	$FFFFCC thru $FFFFC0	RESERVED
PORT C	$FFFFBF	PORT C CONTROL REGISTER (PCRC)
	$FFFFBE	PORT C DIRECTION REGISTER (PRRC)
	$FFFFBD	PORT C GPIO DATA REGISTER (PDRC)

Table C-1. Internal I/O Memory Map (continued)

Peripheral	Address	Register Name
ESAI	$FFFFBC	ESAI RECEIVE SLOT MASK REGISTER B (RSMB)
	$FFFFBB	ESAI RECEIVE SLOT MASK REGISTER A (RSMA)
	$FFFFBA	ESAI TRANSMIT SLOT MASK REGISTER B (TSMB)
	$FFFFB9	ESAI TRANSMIT SLOT MASK REGISTER A (TSMA)
	$FFFFB8	ESAI RECEIVE CLOCK CONTROL REGISTER (RCCR)
	$FFFFB7	ESAI RECEIVE CONTROL REGISTER (RCR)
	$FFFFB6	ESAI TRANSMIT CLOCK CONTROL REGISTER (TCCR)
	$FFFFB5	ESAI TRANSMIT CONTROL REGISTER (TCR)
	$FFFFB4	ESAI COMMON CONTROL REGISTER (SAICR)
	$FFFFB3	ESAI STATUS REGISTER (SAISR)
	$FFFFB2	RESERVED
	$FFFFB1	RESERVED
	$FFFFB0	RESERVED
	$FFFFAF	RESERVED
	$FFFFAE	RESERVED
	$FFFFAD	RESERVED
	$FFFFAC	RESERVED
	$FFFFAB	ESAI RECEIVE DATA REGISTER 3 (RX3)
	$FFFFAA	ESAI RECEIVE DATA REGISTER 2 (RX2)
	$FFFFA9	ESAI RECEIVE DATA REGISTER 1 (RX1)
	$FFFFA8	ESAI RECEIVE DATA REGISTER 0 (RX0)
	$FFFFA7	RESERVED
	$FFFFA6	ESAI TIME SLOT REGISTER (TSR)
	$FFFFA5	ESAI TRANSMIT DATA REGISTER 5 (TX5)
	$FFFFA4	ESAI TRANSMIT DATA REGISTER 4 (TX4)
	$FFFFA3	ESAI TRANSMIT DATA REGISTER 3 (TX3)
	$FFFFA2	ESAI TRANSMIT DATA REGISTER 2 (TX2)
	$FFFFA1	ESAI TRANSMIT DATA REGISTER 1 (TX1)
	$FFFFA0	ESAI TRANSMIT DATA REGISTER 0 (TX0)

Table C-1. Internal I/O Memory Map (continued)

Peripheral	Address	Register Name
Reserved	$FFFF9F thru $FFFF95	RESERVED
SHI	$FFFF94	SHI RECEIVE FIFO (HRX)
	$FFFF93	SHI TRANSMIT REGISTER (HTX)
	$FFFF92	SHI I^2C SLAVE ADDRESS REGISTER (HSAR)
	$FFFF91	SHI CONTROL/STATUS REGISTER (HCSR)
	$FFFF90	SHI CLOCK CONTROL REGISTER (HCKR)
Reserved	$FFFF8F thru $FFFF80	RESERVED

Table C-2. DSP56364 Interrupt Vectors

Interrupt Starting Address	Interrupt Priority Level Range	Interrupt Source
VBA:$00	3	Hardware $\overline{\text{RESET}}$
VBA:$02	3	Stack Error
VBA:$04	3	Illegal Instruction
VBA:$06	3	Debug Request Interrupt
VBA:$08	3	Trap
VBA:$0A	3	Non-Maskable Interrupt ($\overline{\text{NMI}}$)
VBA:$0C	3	Reserved For Future Level-3 Interrupt Source
VBA:$0E	3	Reserved For Future Level-3 Interrupt Source
VBA:$10	0 - 2	IRQA
VBA:$12	0 - 2	IRQB
VBA:$14	0 - 2	Reserved
VBA:$16	0 - 2	IRQD
VBA:$18	0 - 2	DMA Channel 0
VBA:$1A	0 - 2	DMA Channel 1
VBA:$1C	0 - 2	DMA Channel 2
VBA:$1E	0 - 2	DMA Channel 3
VBA:$20	0 - 2	DMA Channel 4
VBA:$22	0 - 2	DMA Channel 5
VBA:$24	0 - 2	Reserved
VBA:$26	0 - 2	Reserved
VBA:$28	0 - 2	Reserved
VBA:$2A	0 - 2	Reserved
VBA:$2C	0 - 2	Reserved

Table C-2. DSP56364 Interrupt Vectors (continued)

Interrupt Starting Address	Interrupt Priority Level Range	Interrupt Source
VBA:$2E	0 - 2	Reserved
VBA:$30	0 - 2	ESAI Receive Data
VBA:$32	0 - 2	ESAI Receive Even Data
VBA:$34	0 - 2	ESAI Receive Data With Exception Status
VBA:$36	0 - 2	ESAI Receive Last Slot
VBA:$38	0 - 2	ESAI Transmit Data
VBA:$3A	0 - 2	ESAI Transmit Even Data
VBA:$3C	0 - 2	ESAI Transmit Data with Exception Status
VBA:$3E	0 - 2	ESAI Transmit Last Slot
VBA:$40	0 - 2	SHI Transmit Data
VBA:$42	0 - 2	SHI Transmit Underrun Error
VBA:$44	0 - 2	SHI Receive FIFO Not Empty
VBA:$46	0 - 2	Reserved
VBA:$48	0 - 2	SHI Receive FIFO Full
VBA:$4A	0 - 2	SHI Receive Overrun Error
VBA:$4C	0 - 2	SHI Bus Error
VBA:$4E	0 - 2	Reserved
:	:	:
VBA:$FE	0 - 2	Reserved

Table C-3. Interrupt Sources Priorities Within an IP

Priority	Interrupt Source
Level 3 (Nonmaskable)	
Highest	Hardware $\overline{\text{RESET}}$
	Stack Error
	Illegal Instruction
	Debug Request Interrupt
	Trap
Lowest	Non-Maskable Interrupt
Levels 0, 1, 2 (Maskable)	
Highest	$\overline{\text{IRQA}}$ (External Interrupt)
	$\overline{\text{IRQB}}$ (External Interrupt)

Table C-3. Interrupt Sources Priorities Within an IP

Priority	Interrupt Source
	$\overline{\text{IRQD}}$ (External Interrupt)
	DMA Channel 0 Interrupt
	DMA Channel 1 Interrupt
	DMA Channel 2 Interrupt
	DMA Channel 3 Interrupt
	DMA Channel 4 Interrupt
	DMA Channel 5 Interrupt
	ESAI Receive Data with Exception Status
	ESAI Receive Even Data
	ESAI Receive Data
	ESAI Receive Last Slot
	ESAI Transmit Data with Exception Status
	ESAI Transmit Last Slot
	ESAI Transmit Even Data
	ESAI Transmit Data
	SHI Bus Error
	SHI Receive Overrun Error
	SHI Transmit Underrun Error
	SHI Receive FIFO Full
	SHI Transmit Data
Lowest	SHI Receive FIFO Not Empty

C.2 Programming Sheets

The following worksheets list the major programmable registers for the DSP56364. The programming sheets are grouped in the following order:

- Central processor,
- Phase Lock Loop, (PLL),
- Enhanced Serial Audio Interface (ESAI),
- Serial Host Interface (SHI),
- GPIO (Ports B-C).

Each sheet provides a space to write in the value of each bit and the hexadecimal value for each register. Programmers can photocopy these sheets and reuse them for each application development project. Refer to the *DSP56300 Family Manual* for further details on the DSP56300 family instruction set.

Application: _____ Date: _____

_____ Programmer: _____

Sheet 1 of 5

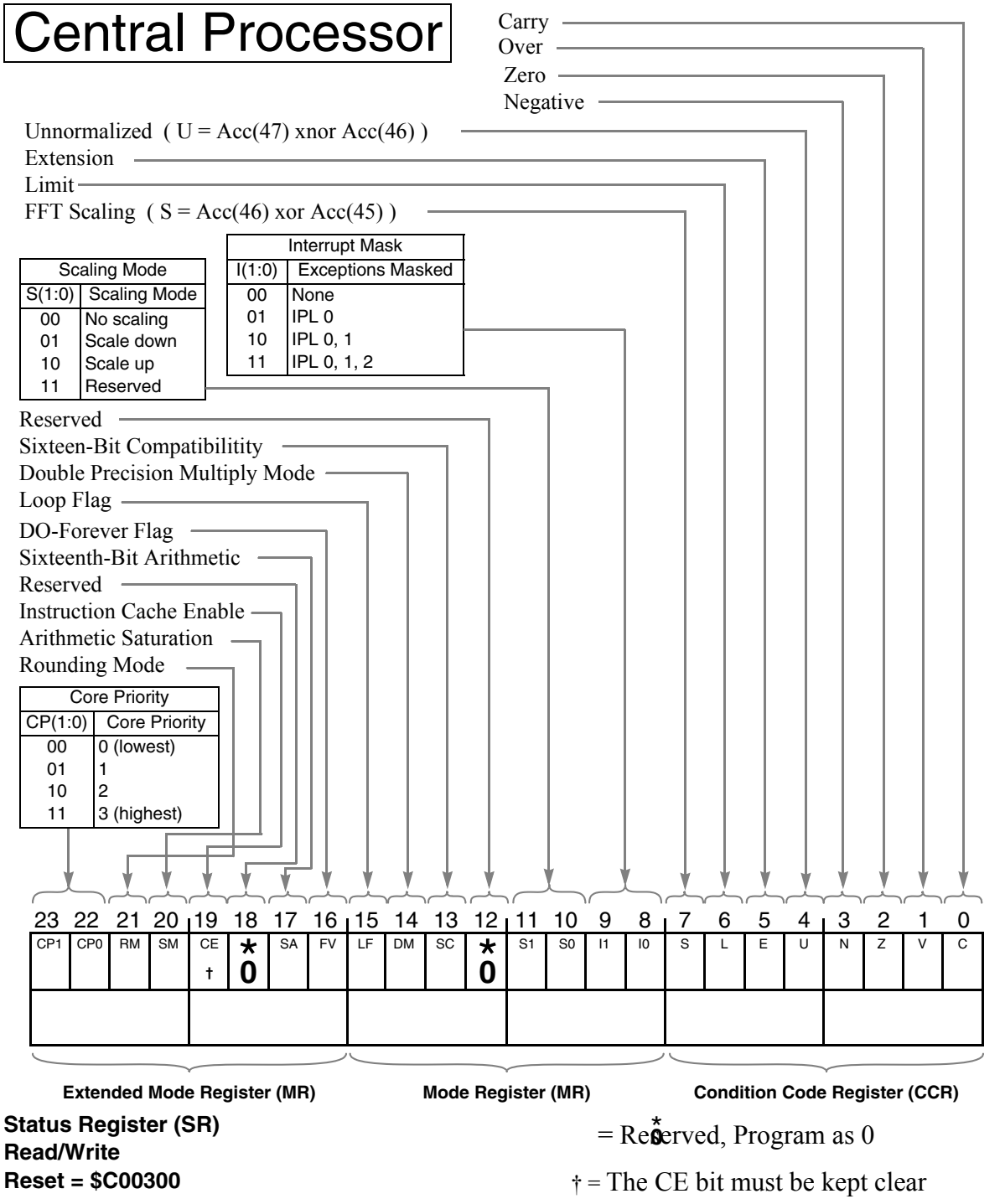


Figure C-1. Status Register (SR)

Application: _____ Date: _____

_____ Programmer: _____

Sheet 2 of 5

Central Processor

Chip Operating Modes		
MOD(D:A)	Reset Vector	Description
(Table 4-1 in Section 4.3, "Operating Modes)		

External Bus Disable
Stop Delay
Memory Switch Mode

Core-DMA Priority	
CDP(1:0)	Core-DMA Priority
00	Core vs DMA Priority
01	DMA accesses > Core
10	DMA accesses = Core
11	DMA accesses < Core

TA Synchronize Select

Address Priority Disable
Address Tracing Enable
Stack Extension Space Select
Extended Stack Underflow Flag
Extended Stack Overflow Flag
Extended Stack Wrap Flag
Stack Extension Enable

23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
*0	*0	*0	SEN	WRP	EOV	EUN	XYS	ATE	APD	*0	*0	TAS	*0	CDP1	CDP0	MS	SD	*0	EBD	MD	*1	MB	MA

System Stack Control Status Register (SCS) | Extended Chip Operating Mode Register (COM) | Chip Operating Mode Register (COM)

Read/Write
Reset = $00030X

*0 = Reserved, Program as 0
*1 =Reserved, program as 1

Figure C-2. Operating Mode Register (OMR)

Application: _____ Date: _____

_____ Programmer: _____

Sheet 3 of 5

CENTRAL PROCESSOR

IRQA Mode

IAL2	Trigger
0	Level
1	Neg. Edge

IAL1	IAL0	Enabled	IPL
0	0	No	—
0	1	Yes	0
1	0	Yes	1
1	1	Yes	2

$\overline{\text{IRQB}}$ Mode

IBL2	Trigger
0	Level
1	Neg. Edge

IBL1	IBL0	Enabled	IPL
0	0	No	—
0	1	Yes	0
1	0	Yes	1
1	1	Yes	2

$\overline{\text{IRQD}}$ Mode

IDL2	Trigger
0	Level
1	Neg. Edge

IDL1	IDL0	Enabled	IPL
0	0	No	—
0	1	Yes	0
1	0	Yes	1
1	1	Yes	2

23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
D5L1	D5L0	D4L1	D4L0	D3L1	D3L0	D2L1	D2L0	D1L1	D1L0	D0L1	D0L0	IDL2	IDL1	IDL0	* 0	* 0	* 0	IBL2	IBL1	IBL0	IAL2	IAL1	IAL0

* **0** = Reserved, Program as 0

Figure C-3. Interrupt Priority Register-Core (IPR-C)

Application: _____ Date: _____

_____ Programmer: _____

Sheet 4 of 5

CENTRAL PROCESSOR

ESAI IPL

ESL1	ESL0	Enabled	IPL
0	0	No	—
0	1	Yes	0
1	0	Yes	1
1	1	Yes	2

SHI IPL

SHL1	SHL0	Enabled	IPL
0	0	No	—
0	1	Yes	0
1	0	Yes	1
1	1	Yes	2

Interrupt Priority Register (IPR–P)
X:$FFFFFE R/W
Reset = $000000

* = Reserved, Program as 0

23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	SHL1	SHL0	ESL1	ESL0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
$0				$0				$0															

Figure C-4. Interrupt Priority Register- Peripherals (IPR-P)

Application: _____ Date: _____

_____ Programmer: _____

Sheet 5 of 5

PLL

PSTP and PEN Relationship					
PSTP	PEN	Operation During STOP		Recovery Time for STOP	Power Consumption during STOP
		PLL	Oscillator		
0	x	Disabled	Disabled	Long	Minimal
1	0	Disabled	Enabled	Short	Lower
1	1	Enabled	Enabled	Short	Higher

Bits XTLR and XTLD
have no effect on
DSP56364 operation

Clock Output Disable (COD)
0 = 50% Duty Cycle Clock
1 = Pin Held In High State

XTAL Disable Bit (XTLD)
0 = Enable Xtal Oscillator
1 = EXTAL Driven From An External Source

Crystal Range Bit (XTLR)
0 = External Xtal Freq > 200KHz
1 = External Xtal Freq < 200KHz

Predivision Factor Bits (PD0 – PD3)	
PD3 – PD0	Predivision Factor PDF
$0	1
$1	2
$2	3
•	•
•	•
•	•
$F	16

Division Factor Bits (DF0 – DF2)	
DF2 – DF0	Division Factor DF
$0	2^0
$1	2^1
$2	2^2
•	•
•	•
•	•
$7	2^7

Multiplication Factor Bits MF0 – MF11	
MF11 – MF0	Multiplication Factor MF
$000	1
$001	2
$002	3
•	•
•	•
•	•
$FFE	4095
$FFF	4096

23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PD3	PD2	PD1	PD0	COD	PEN	PSTP	XTLD	XTLR	DF2	DF1	DF0	MF11	MF10	MF9	MF8	MF7	MF6	MF5	MF4	MF3	MF2	MF1	MF0

PLL Control Register (PCTL)
X:$FFFFFD Read/Write
Reset = $010005

Figure C-5. Phase Lock Loop Control Register (PCTL)

Application:_____ Date:_____

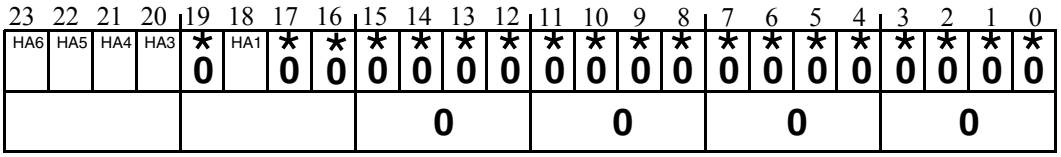
_____ Programmer:_____

Sheet 1 of 3

SHI

SHI Slave Address Register (HSAR)
X:$FFFF92
Reset = $Bx0000

HSAR I^2C Slave Address
Slave address = Bits HA6-HA3, HA1 and external pins HA2, HA0
Slave address after reset = 1011[HA2]0[HA0]



SHI Slave Address Register (HSAR)

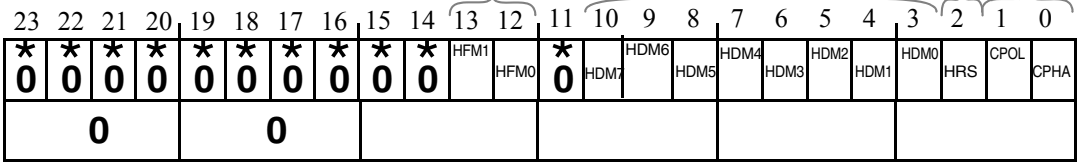
HFM1	HFM0	SHI Noise Reduction Filter Mode
0	0	Bypassed (Filter disabled)
0	1	Reserved
1	0	Narrow spike tolerance
1	1	Wide spike tolerance

CPOL	CPHA	Result
0	0	SCK active low, strobe on rising edge
0	1	SCK active low, strobe on falling edge
1	0	SCK active high, strobe on falling edge
1	1	SCK active high, strobe on rising edge

HRS	Result
0	Prescaler operational
1	Prescaler bypassed

SHI Clock Control Register (HCKR)
X:$FFFF90
Reset = $000001

HCKR Divider Modulus Select



* 0 = Reserved, write as 0

SHI Clock Control Register (HCKR)

Figure C-6. SHI Slave Address (HSAR) and Clock Control Register (HCKR)

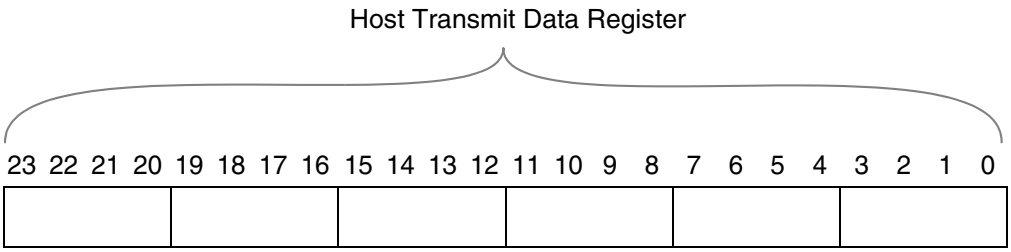
Application: _____ Date: _____

_____ Programmer: _____

Sheet 2 of 3

SHI

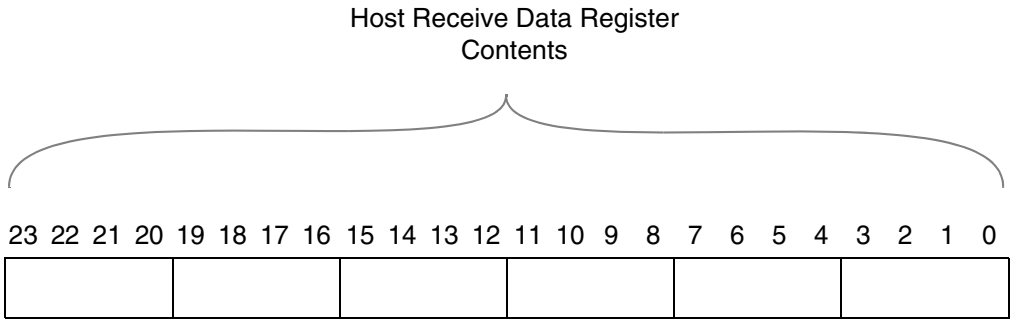
SHI Host Transmit Data Register (HTX) X:$FFFF93 Write Only Reset = $xxxxxx



SHI Host Transmit Data Register (HTX)

SHI Host Receive

Data Register (HRX) X:$FFFF94 Read Only Reset = $xxxxxx



SHI Host Receive Data Register (HRX) (FIFO) 10 words deep

Figure C-7. SHI Host Transmit Data Register (HTX) and Host Receive Data Register (HRX) (FIFO)

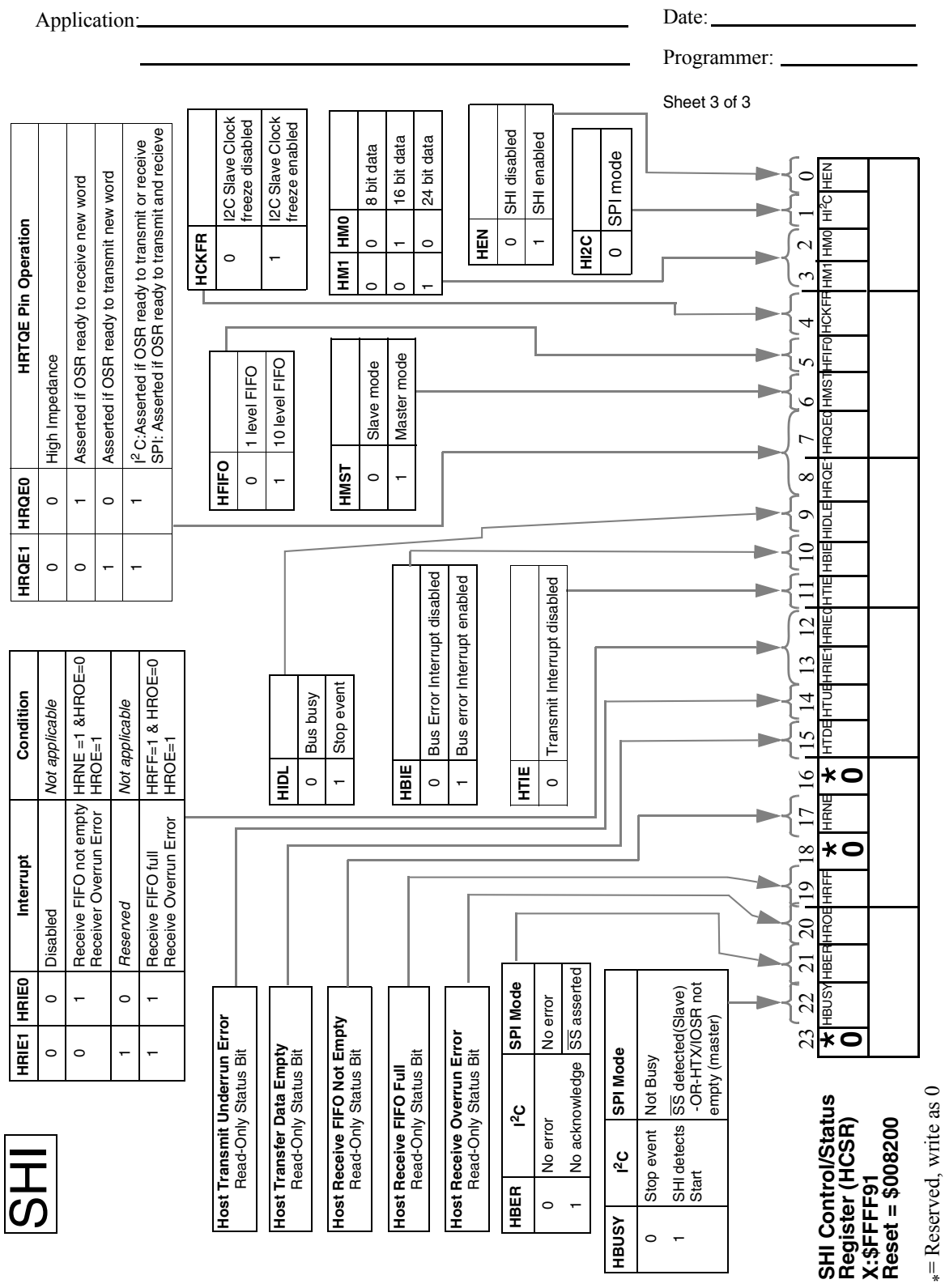


Figure C-8. SHI Host Control/Status Register (HCSR)

Application: _____ Date: _____

Programmer: _____

ESAI

TCCR - ESAI Transmit Clock Control Register
X: $FFFFB6 Reset: $000000

THCKD	
0	HCKT is input
1	HCKT is output

TFSD	
0	FST is input
1	FST is output

TCKD	
0	External clock source used
1	Internal clock source used

THCKP	
0	Transmitter High Frequency Clock Polarity set to clockout on rising edge of transmit clock, latch on falling edge.
1	Transmitter High Frequency Clock Polarity set to clockout on falling edge of transmit clock, latch on rising edge.

TFSP	
0	Frame sync polarity positive
1	Frame sync polarity negative

TCKP	
0	Transmitter Clock Polarity set to clockout on rising edge of transmitter clock, latch in on falling edge of transmit clock
1	Transmitter Clock Polatiry set to clockout on falling edge of transmit clock, latch in on rising edge of transmit clock

TFP [3:0]	Sets divide rate for transmission high frequency clock. Range $0-$F (1-16)

TDC [4:0]	Divider control. Range $00-FF (1-32)

TPSR	
0	Divide by 8 prescaler operational
1	Divide by 8 prescaler bypassed

TPM [7:0]	Specifies the prescaler divide rate is for the transmitter clock generator Range from $00-FF (1-256).

23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
THCKD	TFSD	TCKD	THCKP	TFSP	TCKP	TFP3	TFP2	TFP1	TFP0	TDC4	TDC3	TDC2	TDC1	TDC0	TPSR	TPM7	TPM6	TPM5	TPM4	TPM3	TPM2	TPM1	TPM0

Figure C-9. ESAI Transmit Clock Control Register (TCCR)

Application:_____ Date:_____

_____ Programmer:_____

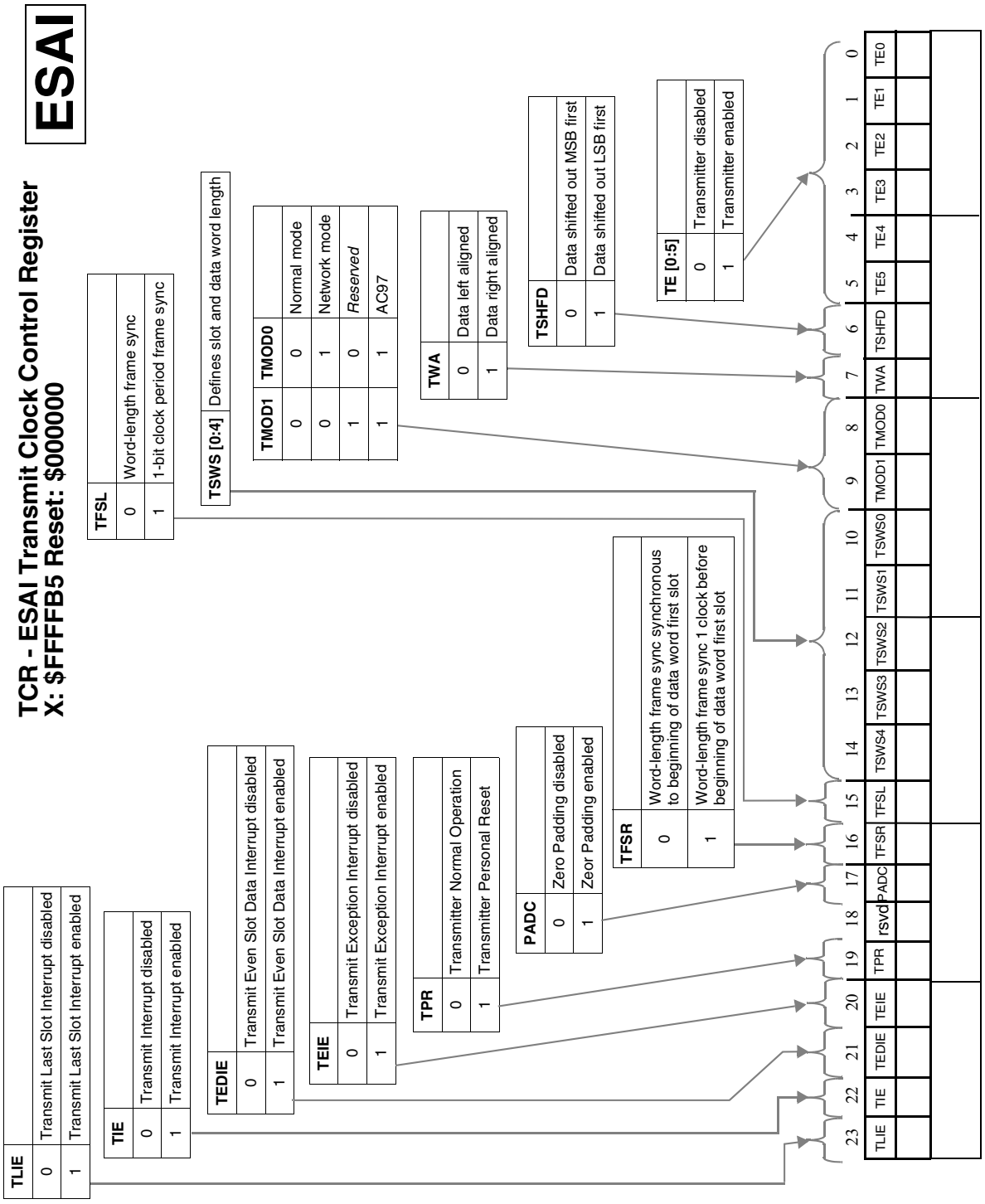


Figure C-10. ESAI Transmit Clock Control Register (TCR)

Application: _____ Date: _____

Programmer: _____

ESAI

RCCR - ESAI Receive Clock Control Register
X: $FFFFB8 Reset: $000000

RHCKD	Description
0	HCKR is input
1	HCKR is output

RFSD	Description
0	FSR is input
1	FSR is output

RCKD	Description
0	External clock source used
1	Internal clock source

RHCKP	Description
0	Clockout on rising edge of receive clock, latch in on falling edge of receive clock
1	Clockout on falling edge of receive clock, latch in on rising edge of receive clock

RFSP	Description
0	Frame sync polarity positive
1	Frame sync polarity negative

RCKP	Description
0	Clockout on rising edge of receive clock, latch in on falling edge of receive clock
1	Clockout on falling edge of receive clock, latch in on rising edge of receive clock

RFP [3:0]	Description
	Sets divide rate for receiver high frequency clock Range $0 - $F (1 -16). See 8.3.3.4

RDC [4:0]	Description
	Controls frame rate dividers Range 00000 - 11111 (1-32) See 8.3.3.2

RPSR	Description
0	Divide by 8 prescaler operational
1	Divide by 8 prescaler bypassed

RPM [7:0]	Description
	Specifies prescaler ratio for the receive clock generator Range from $00 - $FF (1 - 256). See 8.3.3.1

23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RHCKD	RFSD	RCKD	RHCKP	RFSP	RCKP	RFP3	RFP2	RFP1	RFP0	RDC4	RDC3	RDC2	RDC1	RDC0	RPSR	RPM7	RPM6	RPM5	RPM4	RPM3	RPM2	RPM1	RPM0

Figure C-11. ESAI Receive Clock Control Register (RCCR)

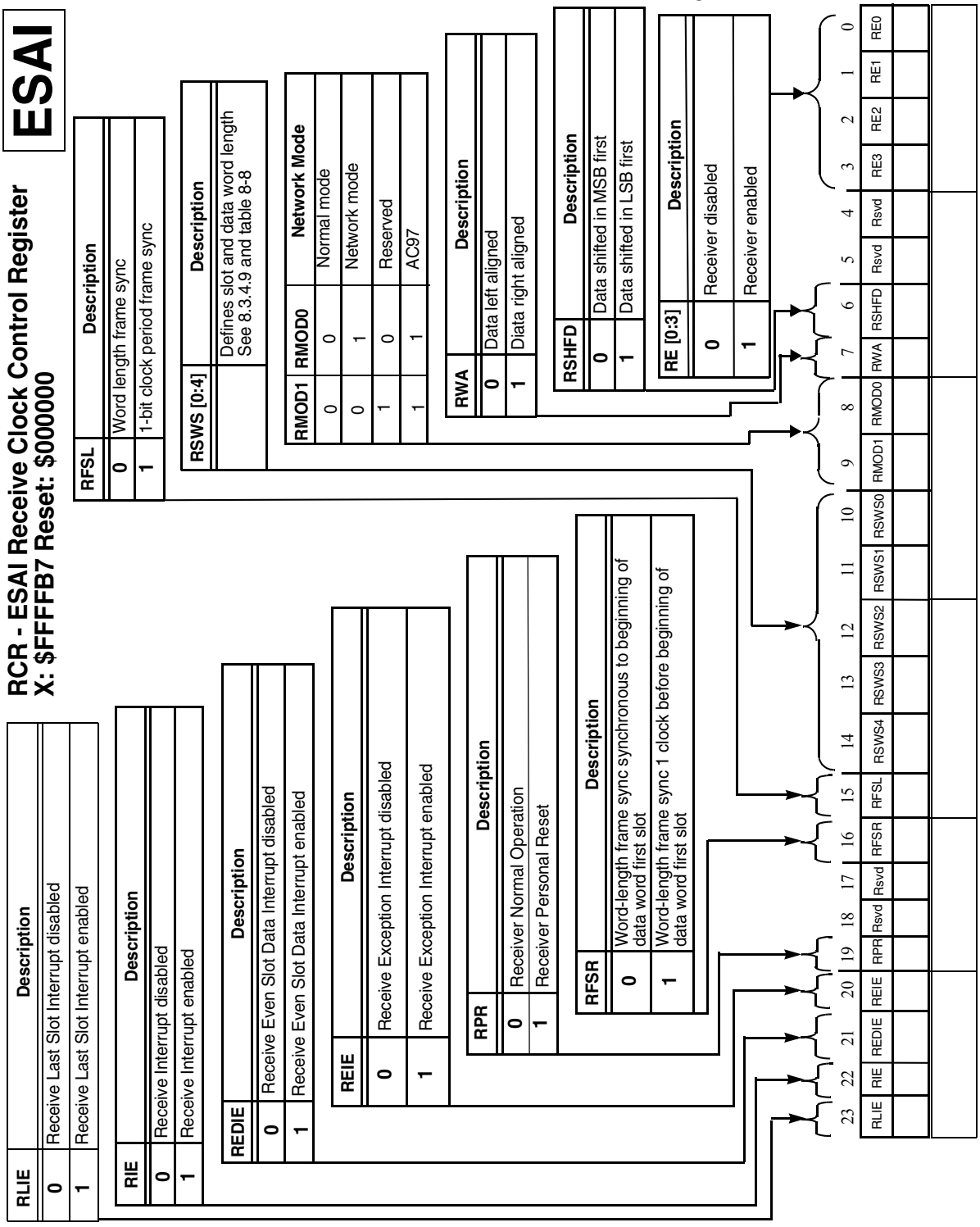


Figure C-12. ESAI Receive Clock Control Register (RCR)

Application: _____ Date: _____

_____ Programmer: _____

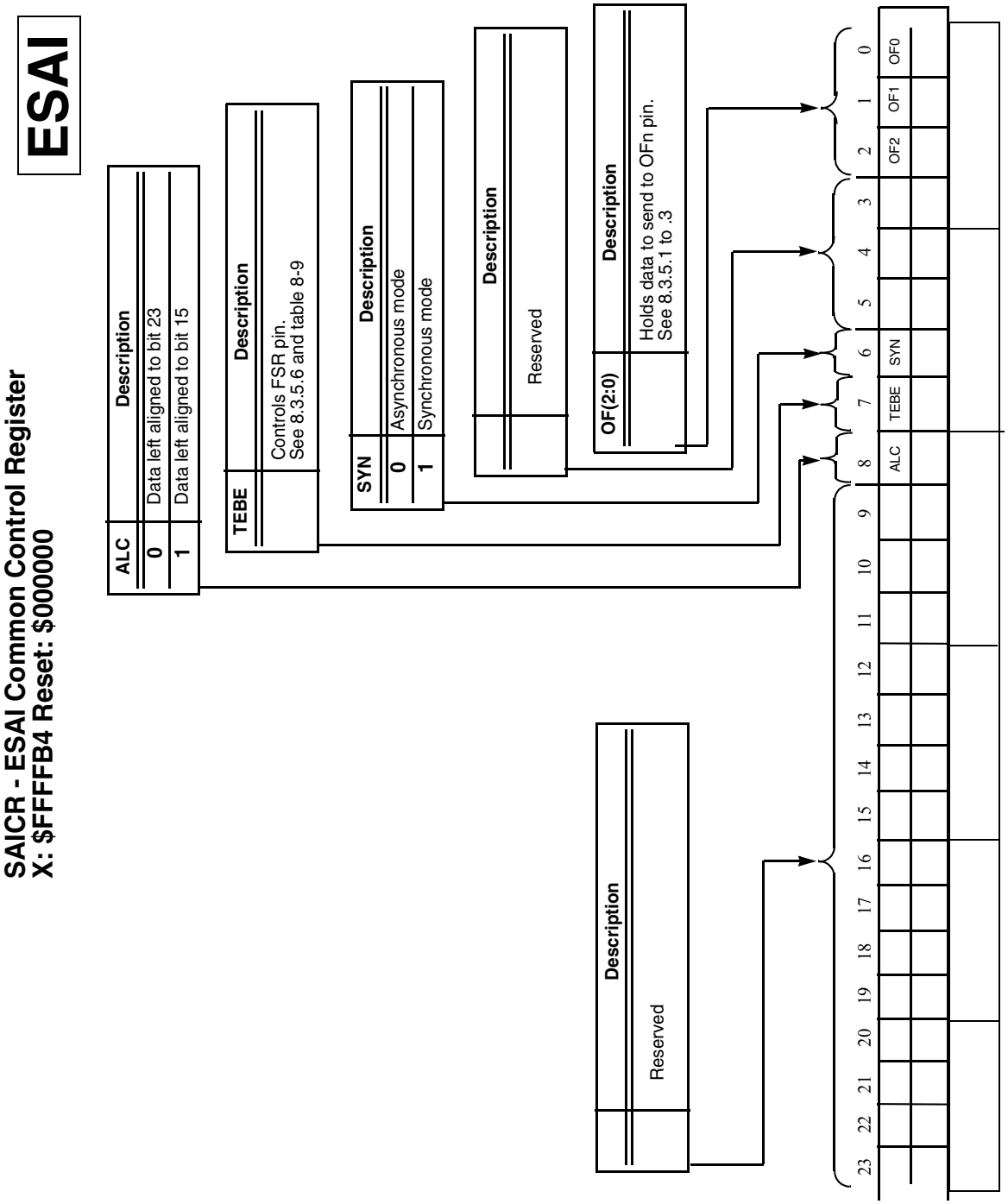


Figure C-13. ESAI Common Control Register (SAICR)

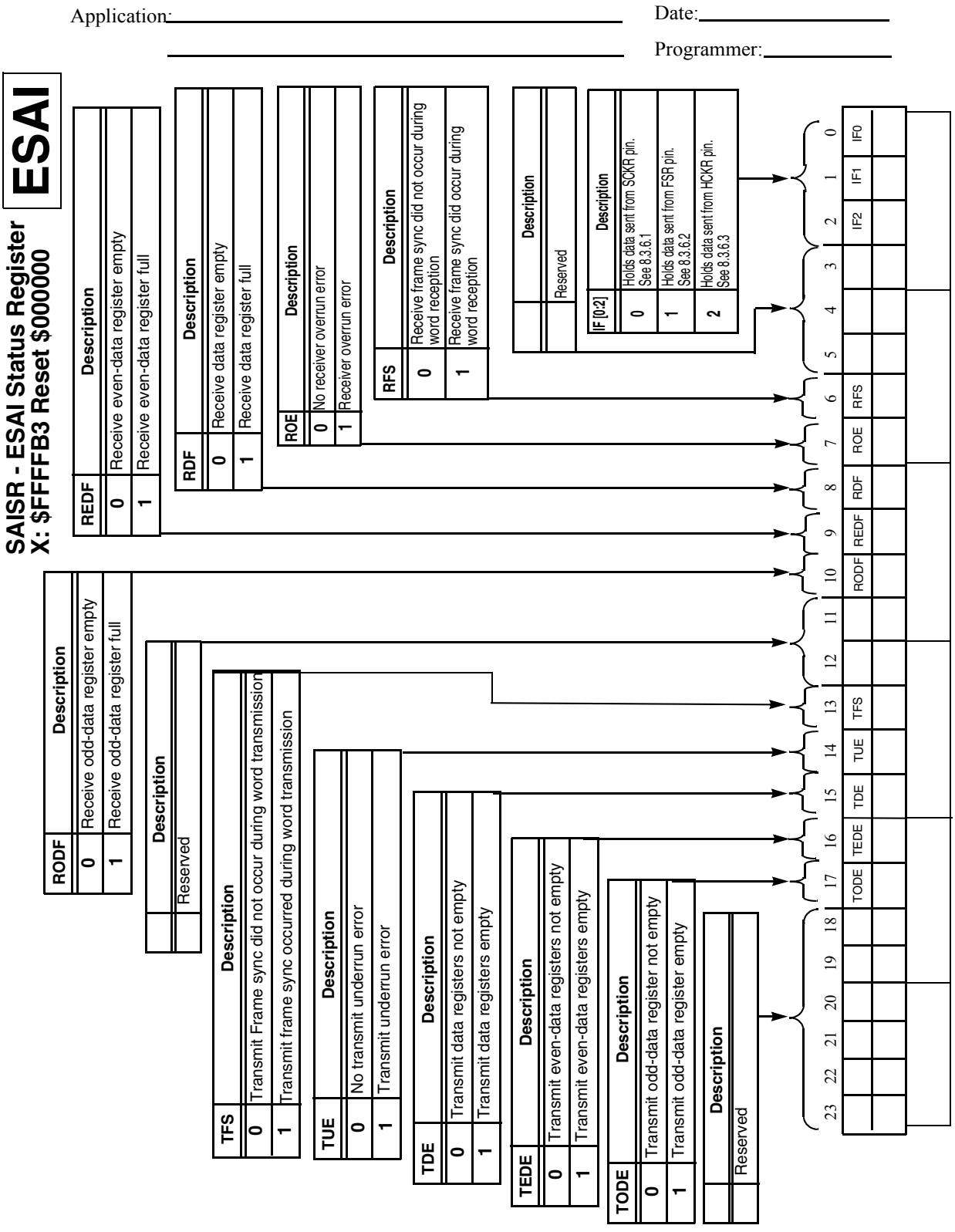


Figure C-14. ESAI Status Register (SAISR)

Application: _____ Date: _____

_____ Programmer: _____

Sheet 2 of 3

GPIO

Port B (GPIO)

Port B Control Register (PCRB)
X:$FFFFCF
ReadWrite
Reset = $0

23 ··· 11	10	9	8	7	6	5	4	3	2	1	0
* 0	* 0	* 0	* 0	* 0	* 0	* 0	* 0	PC3	PC2	PC1	PC0

Port B Direction Register (PRRB)
X:$FFFFCE
ReadWrite
Reset = $0

23 ··· 11	10	9	8	7	6	5	4	3	2	1	0
* 0	* 0	* 0	* 0	* 0	* 0	* 0	* 0	PDC3	PDC2	PDC1	PDC0

PCn = 0 & PDCn = 0 -> Port pin PCn disconnected
PCn = 1 & PDCn = 0 -> Port pin PCn configured as input
PCn = 0 & PDCn = 1 -> Port pin PCn configured as output
PCn = 1 & PDCn = 1 -> Open-Drain Output

Port B GPIO Data Register (PDRB)
X:$FFFFCD
ReadWrite
Reset = $0

23 ··· 11	10	9	8	7	6	5	4	3	2	1	0
* 0	* 0	* 0	* 0	* 0	* 0	* 0	* 0	PD3	PD2	PD1	PD0

If port pin n is GPIO input, then PDn reflects the value on port pin n
if port pin n is GPIO output, then value written to PDn is reflected on port pin n

* = Reserved, Program as 0

Figure C-15. Port B Registers (PCRB, PRRB, PDRB)

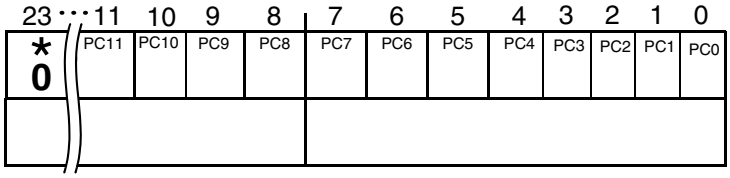
Application: _____ Date: _____

_____ Programmer: _____

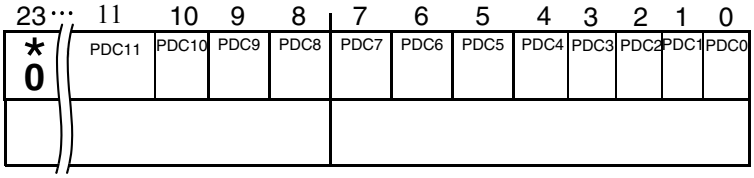
GPIO

Port C (ESAI)

Port C Control Register (PCRC)
X:$FFFFBF
ReadWrite
Reset = $0

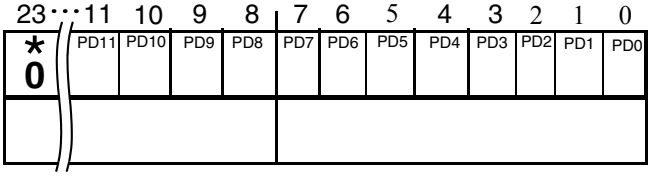


Port C Direction Register (PRRC)
X:$FFFFBE
ReadWrite
Reset = $0



PCn = 0 & PDCn = 0 -> Port pin PCn disconnected
PCn = 1 & PDCn = 0 -> Port pin PCn configured as input
PCn = 0 & PDCn = 1 -> Port pin PCn configured as output
PCn = 1 & PDCn = 1 -> Port pin configured as ESAI

Port C GPIO Data Register (PDRC)
X:$FFFFBD
ReadWrite
Reset = $0



If port pin n is GPIO input, then PDn reflects the value on port pin n
if port pin n is GPIO output, then value written to PDn is reflected on port pin n

* = Reserved, Program as 0

Figure C-16. Port C Registers (PCRC, PRRC, PDRC)

Index

I

J

L

M

O

P

R

S

T

V

X

Y