5685X

User Manual

5685X
Digitial Signal Controller

DSP5685XUM
Rev. 4
07/2005

freescale.com

freescale™
semiconductor

This manual is one of a set of three documents. For complete product information, it is necessary to have all three documents. They are: DSP56800E Reference Manual, DSP5685X User Manual, and Technical Data Sheet.

HOME PAGE: http://www.freescale.com

Order this document as **DSP56F85XUM - Rev 4.0**
June, 2005

Summary of Changes and Updates:

Clarified HI8 Chapter Table 16-1
Converted to Freescale format

TABLE OF CONTENTS

Chapter 1
5685X Overview

Chapter 2
Pin Descriptions

Chapter 3
Memory (MEM)

Chapter 4
System Integration Module (SIM)

Chapter 5
External Memory Interface (EMI)

Chapter 6
On-Chip Clock Synthesis (OCCS)

Chapter 7
Power-On Reset (POR) and Computer Operating Properly (COP)

Chapter 8
Interrupt Controller (ITCN)

Chapter 9
Direct Memory Access (DMA)

Chapter 10
Serial Communications Interface (SCI)

Chapter 11
Serial Peripheral Interface (SPI)

Chapter 12
Enhanced Synchronous Serial Interface (ESSI)

Chapter 13
Quad Timer (TMR)

Chapter 14
Time-Of-Day (TOD)

Chapter 15
General Purpose Input/Output (GPIO)

Chapter 16
Host Interface Eight (HI8)

Chapter 17
JTAG Port

Appendix A
Glossary

Appendix B
Programmer’s Sheets

LIST OF FIGURES

LIST OF TABLES

Preface

About This Manual

Features of the 5685X 16-bit Digital Signal Controllers (DSCs) are described in this manual. Details of memory, operating modes, and peripheral modules are documented here. This manual is intended to be used with the *56800E Reference Manual* (DSP56800ERM), describing the Central Processing Unit (CPU), programming models, and instruction set details. The *DSP56853/+ Technical Data Sheet* provides electrical specifications as well as timing, pinout, and packaging descriptions.

Audience

Information in this manual is intended to assist design and software engineers to integrate a 5685x device into a design and/or while developing application software.

Manual Organization

This manual is arranged in sections described below:

- Chapter 1, 5685X Overview—provides a brief overview, describing the structure of this document, including lists of other documentation necessary to use these chips.
- Chapter 2, Pin Descriptions—describes package pins for each device and how the pins are grouped into the various interfaces.
- Chapter 3, Memory (MEM)—depicts the on-chip memory, structures, registers, and interfaces.
- Chapter 4, System Integration Module (SIM)—documents the system control functions module.
- Chapter 5, External Memory Interface (EMI)—defines specifications for the IPBus based External Memory Interface (EMI). The EMI module is used for each of the 56800E Family chips contained in this manual.
- Chapter 6, On-Chip Clock Synthesis (OCCS)—elaborates on the internal oscillator, Phase Lock Loop (PLL), and timer distribution chain for the 5685x.

- **Chapter 7, Power-On Reset (POR) and Computer Operating Properly (COP)**—function monitors the core power supply and analog power supply.
- Chapter 8, Interrupt Controller (ITCN)—describes how the IPBus Interrupt Controller accepts interrupt requests from IPBus-based peripherals and presents them to the 56800E core.
- Chapter 9, Direct Memory Access (DMA)—transfers data between points in the memory map without intervention by the CPU. The DMA controller allows movements of data to and from internal data memory, or internal peripherals to occur in the background of CPU operation.
- **Chapter 10, Serial Communications Interface (SCI)**—presents the Serial Communications Interface, communicating with devices such as codecs, other DSCs, microprocessors, and peripherals to provide the primary data input path.
- **Chapter 11, Serial Peripheral Interface (SPI)**—describes the Serial Peripheral Interface, which communicates with external devices, such as Liquid Crystal Displays (LCDs) and Microcontroller Units (MCUs).
- Chapter 12, Enhanced Synchronous Serial Interface (ESSI)—details the Enhanced Synchronous Serial Interface. It communicates with devices such as industry-standard codecs, other DSCs, microprocessors, and peripherals to implementing the Serial Peripheral Interface (SPI).
- Chapter 13, Quad Timer (TMR)—outlines the internal Quad Timer devices available, including features and registers.
- **Chapter 14, Time-Of-Day (TOD)**—discusses instruction of the sequence of counters to track elapsed time and its ability to track time up to 179.5 years, or 65,535 days.
- Chapter 15, General Purpose Input/Output (GPIO)—describes how peripheral pins are multiplexed with GPIO functions.
- Chapter 16, Host Interface Eight (HI8)—provides host processors with DSC design accesses.
- **Chapter 17, JTAG Port**—explains the Joint Test Action Group (JTAG) testing methodology and its capabilities with Test Access Port (TAP) and Enhanced OnCE (fully explained in the Reference Manual).
- **Appendix A, Glossary**—provides definitions of terms, peripherals, acronyms and register names used in this manual.
- Appendix B, Programmer's Sheets—provides concise, one location of registers and their reference tables intended to simplify programming of the 5685x.

Suggested Reading

A list of related books is provided here as an aid those who may be new to Digital Signal Controllers:

Advanced Topics in Signal Processing, Jae S. Lim and Alan V. Oppenheim (Prentice-Hall: 1988).

Applications of Digital Signal Processing, A. V. Oppenheim (Prentice-Hall: 1978).

Digital Processing of Signals: Theory and Practice, Maurice Bellanger (John Wiley and Sons: 1984).

Digital Signal Processing, Alan V. Oppenheim and Ronald W. Schafer (Prentice-Hall: 1975).

Digital Signal Processing: A System Design Approach, David J. DeFatta, Joseph G. Lucas, and William S. Hodgkiss (John Wiley and Sons: 1988).

Discrete-Time Signal Processing, A. V. Oppenheim and R.W. Schafer (Prentice-Hall: 1989).

Foundations of Digital Signal Processing and Data Analysis, J. A. Cadzow (Macmillan: 1987).

Handbook of Digital Signal Processing, D. F. Elliott (Academic Press: 1987).

Introduction to Digital Signal Processing, John G. Proakis and Dimitris G. Manolakis (Macmillan: 1988).

IP Bus Specifications, Semiconductor Reuse Standard, SRSIPB1, v 2.0, Draft 1.6.

Multirate Digital Signal Processing, R. E. Crochiere and L. R. Rabiner (Prentice-Hall: 1983).

Signal Processing Algorithms, S. Stearns and R. Davis (Prentice-Hall: 1988).

Signal Processing Handbook, C. H. Chen (Marcel Dekker: 1988).

Signal Processing: The Modern Approach, James V. Candy (McGraw-Hill: 1988).

Theory and Application of Digital Signal Processing, Lawrence R. Rabiner and Bernard Gold (Prentice-Hall: 1975).

Manual Conventions

Conventions used in this manual:

- Bits within registers are always listed from Most Significant Bit (MSB) to Least Significant Bit (LSB).
- Bits within a register are formatted AA[n:0] when more than one bit is involved in a description. 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 set to *one*. When a bit is described as *cleared*, its value is set to *zero*.
- Pins or signals asserted low, made active when pulled to ground, have an over-bar above their name. For example, the $\overline{SS0}$ pin is asserted low.
- Hex values are indicated with a dollar sign ($) preceding the hex value, as follows: $FFFB is the X memory address for the Interrupt Priority Register (IPR).
- Code examples follow in a single spaced font.

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

                    ; MISO0, MOSI0, SCK0 for SPI master       line 2

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

- Pins or signals listed in code examples asserted as low have a tilde in front of their names. In the previous example, line three refers to the $\overline{SS0}$ pin, shown as ~SS0.
- The word *reset* is used in three different contexts in this manual. The word *pin* is a generic term for any pin on the chip. They are described as:
  - a reset pin is always written as $\overline{RESET}$, in uppercase, using the over bar
  - the processor state occurs when the $\overline{RESET}$ pin is asserted. It is always written as Reset, with a capitalized first letter
  - the word *reset* refers to the reset function. It is written in lowercase, without italics, used here only for differentiation. The word may require a capital letter as style dictates, such as in headings and captions
- The word *assert* means a high true (active high) signal is pulled high to V_{DD}, or a low true (active low) signal is pulled low to ground. The word *deassert* means a high true signal is pulled low to ground, or a low true signal is pulled high to V_{DD}.

Table 0-1. Pin Conventions

Signal/Symbol	Logic State	Signal State	Voltage[1]
$\overline{\text{PIN}}$	True	Asserted	V_{IL}/V_{OL}
$\overline{\text{PIN}}$	False	Deasserted	V_{IH}/V_{OH}
PIN	True	Asserted	V_{IH}/V_{OH}
PIN	False	Deasserted	V_{IL}/V_{OL}

1.Values for VIL, VOL, VIH, and VOH are defined by individual product specifications.

The following standards are recognized in choosing block I/O signal names for the bridge:

- For clarity, all signals are referenced with capital letters throughout this document
- Descriptive functionality of signals is taken into account when choosing signal names. This may imply names for system bus interface signals that are different than those defined in the 56800E specification documents.
- All efforts are made to maintain compliance with Semiconductor Reuse Standards guidelines
- The prefix *IPBB_* is used for all signals initiated from the IP Bus bridge
- A prefix symbolizing a specific block's name is used to distinguish input signals (point to point signals)
- All signals initiated from a particular bus will contain a prefix signifying that bus

Throughout the manual, registers displaying a grayed area designate reserved bits.

= Reserved or not implemented bit, written as zero for future compatibility

Chapter 1
5685X Overview

1.1 Introduction

Differing in size of memories and choice of peripherals, these multi-functional chips offer features DSP programmers demand for executing true signal processing algorithms, flexible user-defined, multi-level interrupt priority support, supports traditional DSP mathematical functions for executing complicated computations, giving programmers more control in an interrupt-driven application.

The 56853, 56854, 56855, 56857, and 56858 are members of the 56800E core-based family of Digital Signal Controllers (DSCs). Combined on a single chip are the processing power of a DSP and the functionality of a microcontroller with a flexible set of peripherals. The chip design creates an extremely cost-effective and compact solution for a number of uses.

The 56800E family includes many peripherals especially well-suited for low-end Internet appliance applications and low-end client applications, such as:

- Telephony
- Voice Recognition
- Voice-Processing for Multi-Processor Systems
- Portable Devices
- Advanced Industrial Control
- Internet Audio
- Point-of-Sale Systems
- Noise suppression
- ID tag readers
- Sonic/Subsonic detectors
- Security access devices
- Remote metering
- Sonic alarms

1.2 5685x Family Description

The 5685x device family is based on the 56800E core and combines, on a single chip, the processing power of a digital signal controller and the functionality of a microcontroller with a flexible set of peripherals to create an extremely cost-effective solution. Because of its low cost, configuration flexibility, and compact program code, each of the five devices are well-suited for many applications.

With the exception of the 56852 chip (not included in this manual), the 5685x family includes from 12K to 40K words of Program SRAM, 4K to 24K words of Data RAM and 1K of Boot ROM.

With the exception of the 56857, the balance of the 5685x family of devices supports program execution from either internal or external memories. Two data operands can be accessed from the on-chip Data RAM per instruction cycle. The 5685x family also provides two external dedicated interrupt lines, and up to 47-General Purpose Input/Output (GPIO) lines, depending on peripheral configuration.

All devices, except the 857, support program execution from external memory. The 56800E core can access two data operands from the on-chip Data RAM per instruction cycle.

These controllers also provides a full set of standard programmable peripherals that include an 8-bit Host Interface (HI8), up to two Enhanced Synchronous Serial Interfaces (ESSI), one Serial Peripheral Interface (SPI), two Serial Communications Interfaces (SCI), and one Quad Timer (TMR). When primary functions of each of the peripheries (HI8, TMR, ESSI, SPI, SCI I/O and four chip selects) are not required, they can be used as General Purpose Input/Outputs (GPIOs). Please see **Section 1.3** for details on each of the 5685x family of devices.

1.2.1 5685x Key Family Features

1.2.1.1 The 56853

- Supports program execution from either internal or external memories
- Provides two external dedicated interrupt lines
- Provides up to 41-General Purpose Input/Output (GPIO) lines, depending on peripheral configuration
- Includes 12K words of Program RAM
- Includes 4K words of Data RAM
- Includes 1K words of Boot ROM
- Provides a full set of standard programmable peripherals including:
  - eight-bit Host Interface (HI8)
  - one Enhanced Synchronous Serial Interface (ESSI)
  - one Serial Peripheral Interface (SPI)
  - two Serial Communications Interfaces (SCIs)
  - one Quad Timer (TMR)
  - one, 6-channel Direct Memory Access (DMA)

The HI8, ESSI, SPI, SCI, four chip selects and TMR can be used as General Purpose Input/Outputs (GPIOs) if its primary function is not required.

1.2.1.2 The 56854

- Supports program execution from either internal or external memories
- Provides two external dedicated interrupt lines
- Provides up to 41-General Purpose Input/Output (GPIO) lines, depending on peripheral configuration
- Includes 16K words of Program RAM
- Includes 16K words of Data RAM
- Includes 1K words of Boot ROM
- Provides a full set of standard programmable peripherals including:
  - eight-bit Host Interface (HI8)
  - one Enhanced Synchronous Serial Interface (ESSI)
  - one Serial Peripheral Interface (SPI)
  - two Serial Communications Interfaces (SCIs)
  - one Quad Timer (TMR)
  - one, 6-channel Direct Memory Access (DMA)

The HI8, ESSI, SPI, SCI, four chip selects and TMR can be used as General Purpose Input/Outputs (GPIOs) if its primary function is not required.

1.2.1.3 The 56855

- Supports program execution from either internal or external memories
- Provides two external dedicated interrupt lines
- Provides up to 18-General Purpose Input/Output (GPIO) lines, depending on peripheral configuration
- Includes 24K words of Program RAM
- Includes 24K words of Data RAM
- Includes 1K of Boot ROM
- Provides a full set of standard programmable peripherals including:
  - one Enhanced Synchronous Serial Interface (ESSI)
  - two Serial Communications Interfaces (SCI)
  - one Quad Timer (TMR)
  - one, 6-channel Direct Memory Access (DMA)

The ESSI, SCI I/O, four chip selects and quad timer can be used as General Purpose Input/Outputs when its primary function is not required.

1.2.1.4 The 56857

- Provides two external dedicated interrupt lines
- Provides up to 47-General Purpose Input/Output (GPIO) lines, depending on peripheral configuration.
- Includes 40K words of Program RAM
- Includes 24K words of Data RAM
- Includes 1K of Boot ROM
- Provides a full set of standard programmable peripherals including:
  - eight-bit Host Interface (HI8)
  - two Enhanced Synchronous Serial Interface (ESSI)
  - one Serial Peripheral Interface (SPI)
  - two Serial Communications Interfaces (SCI)
  - one Quad Timer (TMR)
  - one, 6-channel Direct Memory Access (DMA)

The HI8, ESSIs, SPI, SCIs I/O and TMR can be used as General Purpose Input/Outputs when its primary function is not required.

1.2.1.5 The 56858

- Supports program execution from either internal or external memories
- Provides two external dedicated interrupt lines
- Provides up to 47-General Purpose Input/Output (GPIO) lines, depending on peripheral configuration
- Includes 40K words of Program RAM
- Includes 24K words of Data RAM
- Includes 1K of Boot ROM
- Provides a full set of standard programmable peripherals that include:
  - eight-bit Host Interface (HI8)
  - two Enhanced Synchronous Serial Interfaces (ESSI)
  - one Serial Peripheral Interface (SPI)
  - two Serial Communications Interfaces (SCI)
  - one Quad Timer (TMR)
  - one, 6-channel Direct Memory Access (DMA)

The HI8, TMR, ESSIs, SPI, SCI’s I/O and four chip selects can be used as General Purpose Input/Outputs when its primary function is not required.

1.3 5685x Family Architectural Overview

The 5685x family of devices consists of the 56800E core, program and data memory, including peripherals useful for embedded control applications. Block diagrams for each chip describing the differences in available peripheral sets and memory shown in the following figures: **Figure 1-1**, **Figure 1-2**, **Figure 1-3**, **Figure 1-4**, and **Figure 1-5**.

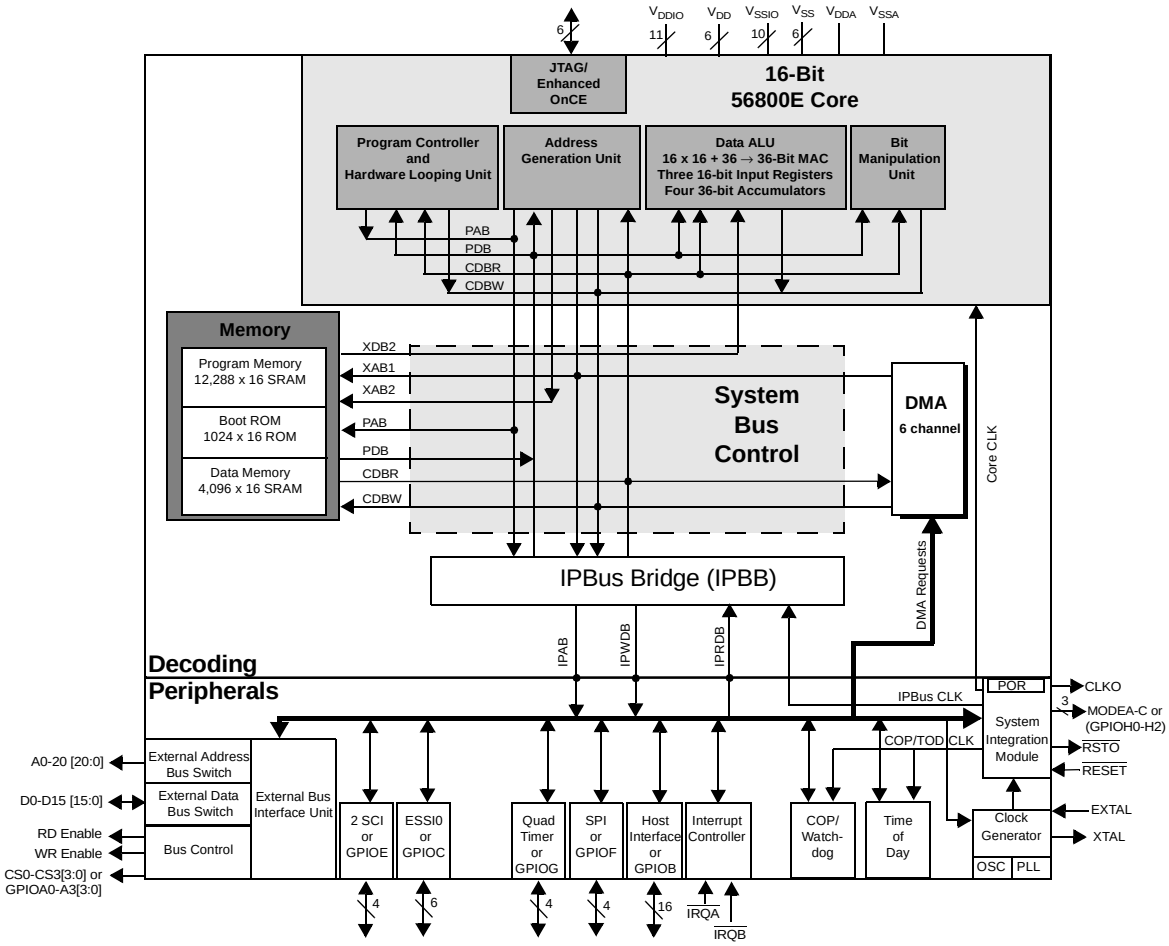


Figure 1-1. 56853 Functional Block Diagram

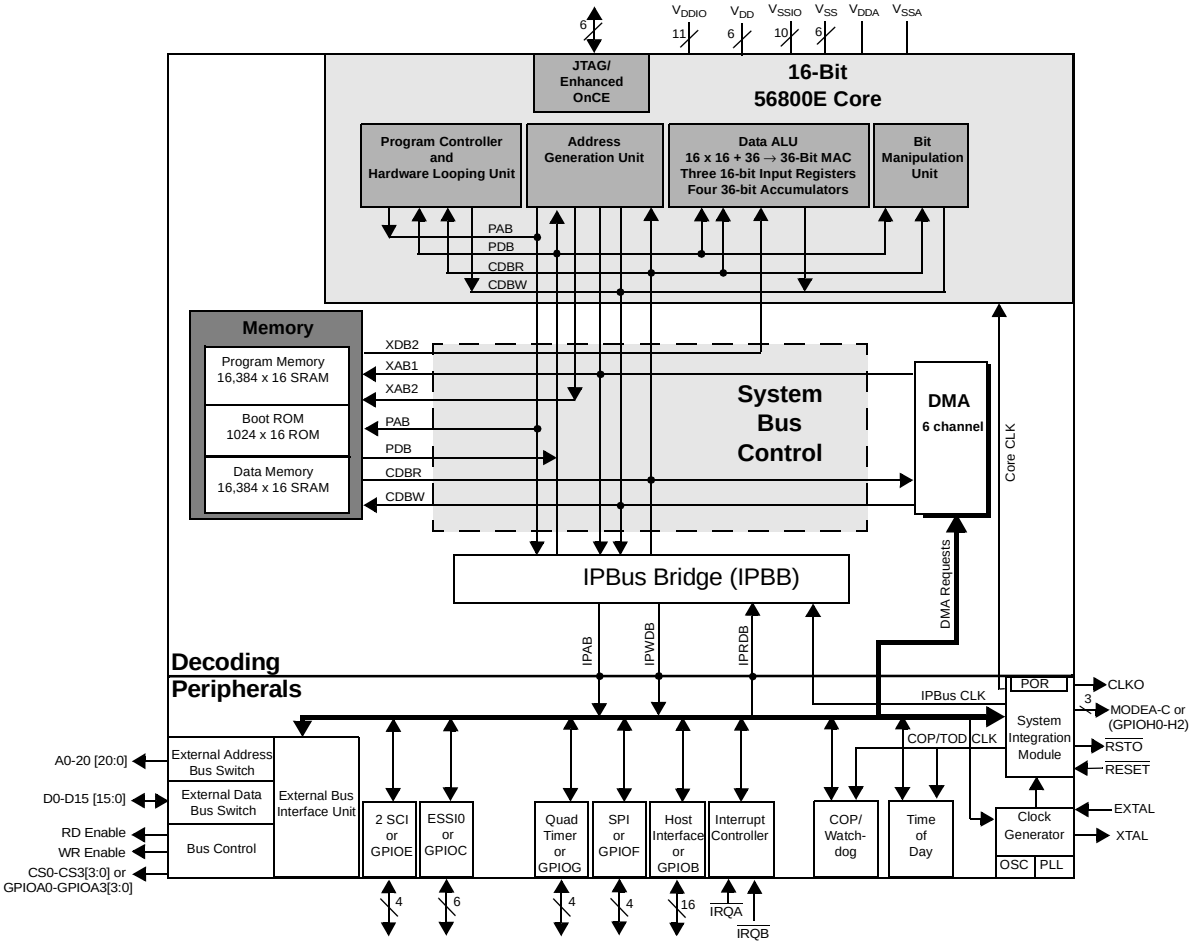


Figure 1-2. 56854 Functional Block Diagram

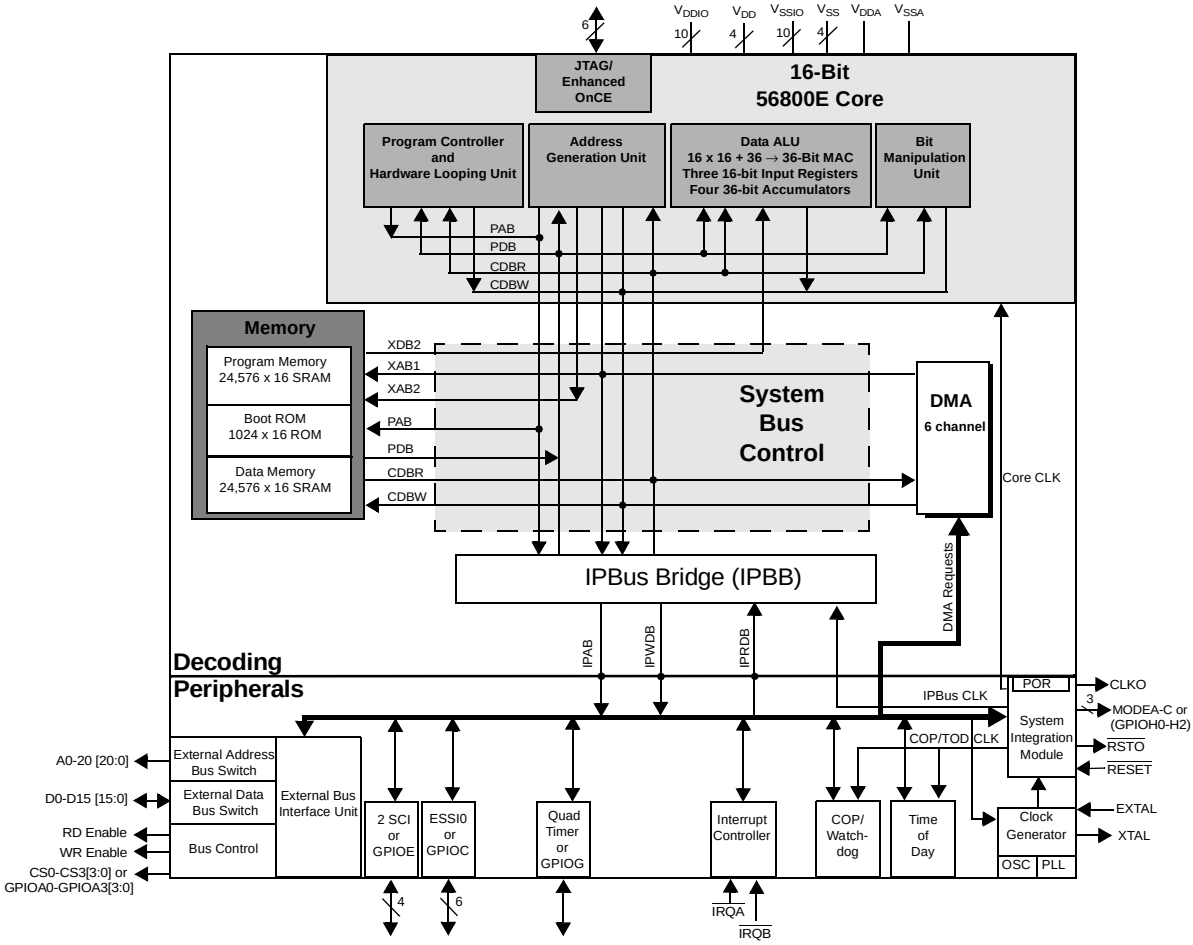


Figure 1-3. 56855 Functional Block Diagram

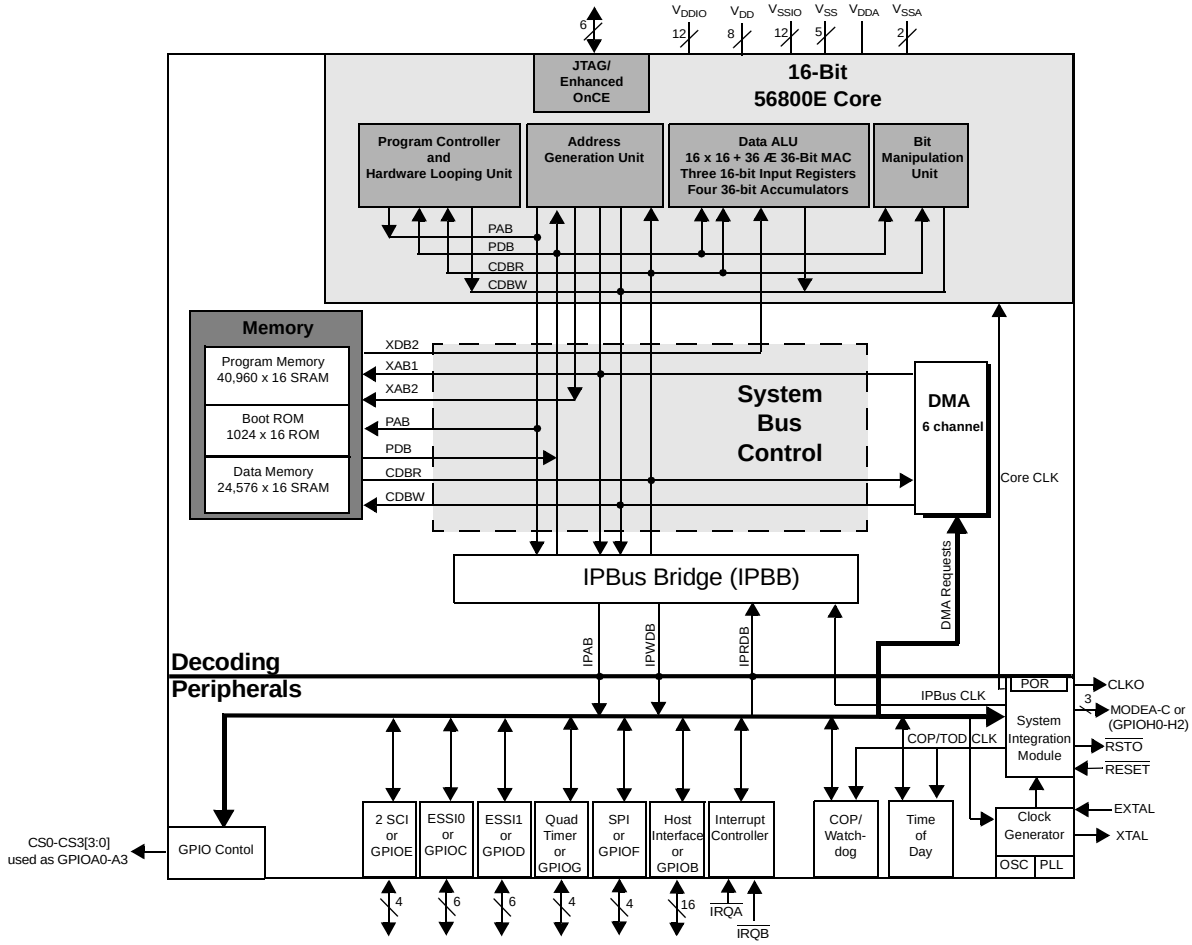


Figure 1-4. 56857 Functional Block Diagram

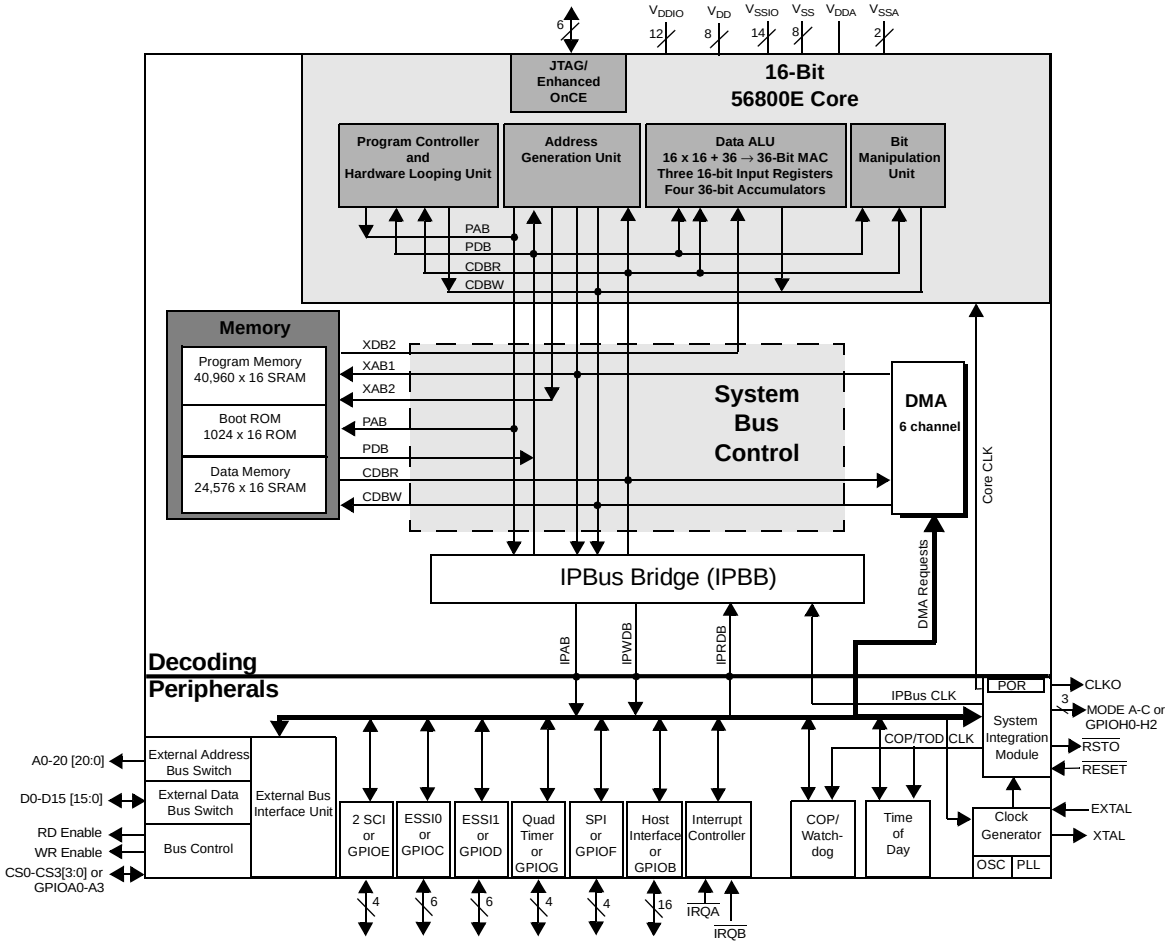


Figure 1-5. 56858 Functional Block Diagram

Table 1-1. Feature Matrix

Feature	56853	56854	56855	56857	56858
Speed (MIPS)	120	120	120	120	120
Program SRAM	12K x 16	16K x 16	24K x 16	40K x 16	40K x 16
Data SRAM	4K x 16	16K x 16	24K x 16	24K x 16	24K x 16
Boot ROM	1K x 16	1K x 16	1K x 16	1K x 16	1K x 16
Program Memory	Up to 2M words	Up to 2M words	Up to 2M words	—	Up to 2M words
Data Memory	Up to 8M words	Up to 8M words	Up to 8M words	—-	Up to 8M words
Oscillator	Yes	Yes	Yes	Yes	Yes
PLL	Yes	Yes	Yes	Yes	Yes
SCI	2	2	2	2	2
SPI	1	1	0	1	1
ESSI	1	1	1	2	2
HI8	1	1	0	1	1
Watchdog	1	1	1	1	1
General Purpose Timers	4	4	4	4	4
Dedicated GPIO	—	—	—	4	—
GPIO (Max)	41	41	18	47	47
JTAG/EOnCE	1	1	1	1	1
Interrupt Controller	Yes	Yes	Yes	Yes	Yes
DMA (6- channel)	Yes	Yes	Yes	Yes	Yes
External Bus	1	1	1	—	1
Package	128 LQFP	128 LQFP	100 LQFP	100 LQFP	144 LQFP 144MAPBGA

1.4 56800E Core Description

This section provides a brief overview of the 56800E core. For a more thorough description, please refer to the *56800E Core Reference Manual* (DSP56800ERM).

1.4.1 Key Features

The 56800E architecture provides a variety of features to enhance performance, reduce application cost, and ease product development. The architectural features making these benefits possible include:

- Efficient 16-bit engine with dual Harvard architecture
- 120 Million Instructions Per Second (MIPS) at 120MHz core frequency
- Single-cycle 16 × 16-bit parallel Multiplier-Accumulator (MAC)
- Four, 36-bit accumulators including extension bits
- 16-bit bidirectional shifter
- Parallel instruction set with unique addressing modes
- Hardware DO and REP loops
- Three internal address buses and one external address bus
- Four internal data buses and one external data bus
- Instruction set supports both digital signal controller and controller functions
- Four hardware interrupt levels
- Five software interrupt levels
- Controller-style addressing modes and instructions for compact code
- Efficient C Compiler and local variable support
- Software subroutine and interrupt stack with depth limited only by memory
- JTAG/Enhanced OnCE debug programming interface

1.4.2 56800E Core Enhancements

The 56800E core architecture extends the 56800 family architecture. It is source-code compatible with 56800 devices and adds the following new features:

- Byte and long data types, supplementing the 56800’s word data type
- 24-bit data memory address space
- 21-bit program memory address space
- Two additional 24-bit pointer registers
- Two additional 36-bit accumulator registers

- Full-precision integer multiplication
- 32-bit logical and shifting operations
- Second read in dual read instruction can access off-chip memory
- Loop Count (LC) register extended to 16 bits
- Support for nested DO looping through additional loop address and count registers
- Loop address and hardware stack extended to 24 bits
- Three additional interrupt levels with a software interrupt for each level
- Enhanced On-Chip Emulation (EOnCE) with three debugging modes:
  - non-intrusive real-time debugging
  - minimally intrusive real-time debugging
  - break point and step modes (core is halted)

1.4.3 System Architecture and Peripheral Interface

The 56800E system architecture encompasses all the on-chip components, including the core, on-chip memory, peripherals, and buses necessary to connect them. **Figure 1-6** shows the overall system architecture for a device with an external bus.

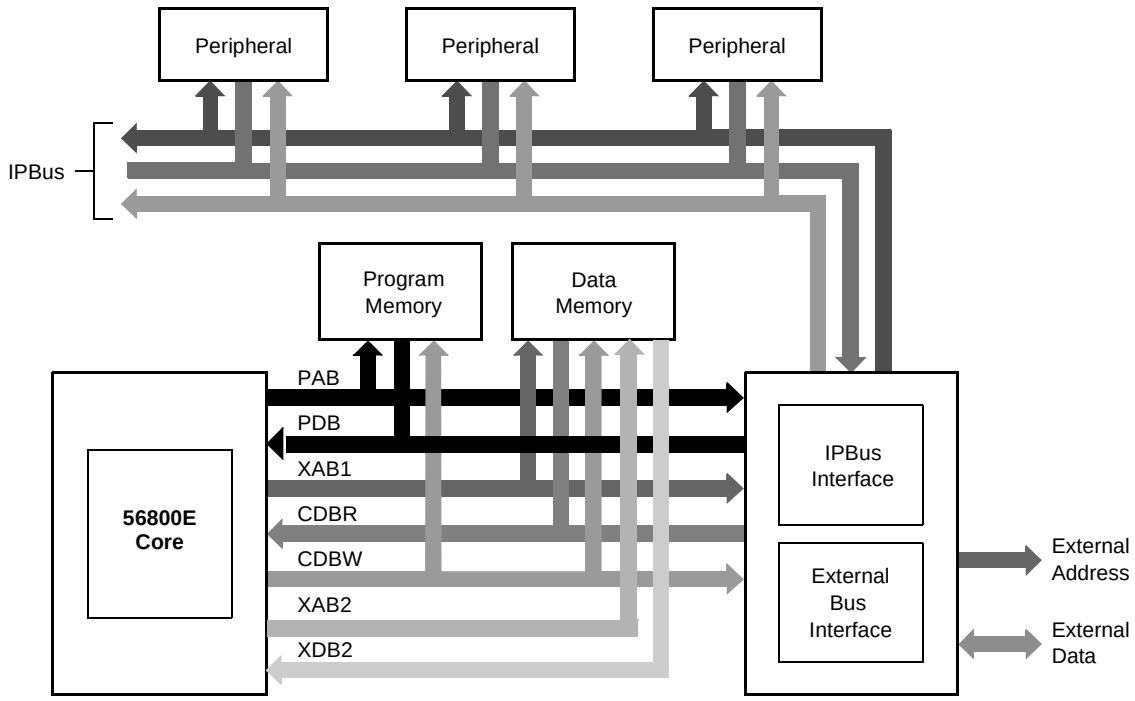


Figure 1-6. 56800E Chip Architecture with External Bus

Complete architecture includes these components:

- 56800E core
- On-chip program memory
- On-chip data memory
- On-chip peripherals
- IPBus peripheral interface
- External bus interface

Some 56800E devices might not implement an external bus interface. Regardless of the implementation, all peripherals communicate with the 56800E core via the IPBus interface. The IPBus interface standard connects the main data address bus XAB1, CDBR, and CDBW unidirectional data buses to the corresponding bus interfaces on the peripheral devices. The program memory buses are not connected to peripherals.

1.4.4 56800E Core Block Diagram

The 56800E core is composed of several independent functional units. The program controller, Address Generation Unit (AGU), and data Arithmetic Logic Unit (ALU) contain their own register sets and control logic, allowing them to operate independently and in parallel, which increases throughput. There is also an independent bit-manipulation unit enabling efficient bit-manipulation operations. Each functional unit interfaces with the other units, memory, and the memory-mapped peripherals over the core's internal address and data buses. A block diagram of the 56800E core architecture is shown in **Figure 1-7.**

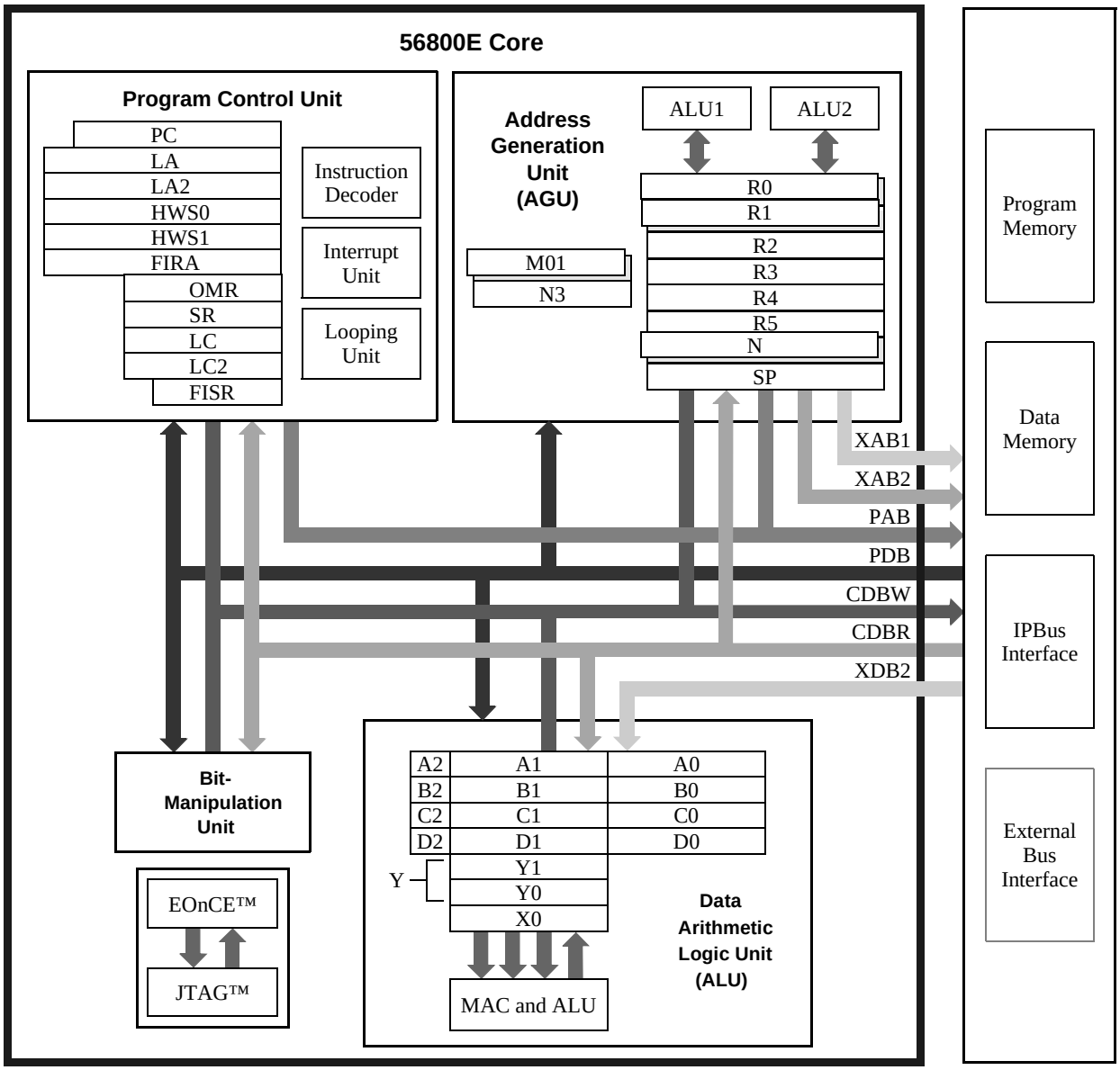


Figure 1-7. 56800E Core Block Diagram

Instruction execution is pipelined to take advantage of the parallel units, significantly decreasing the execution time for each instruction. For example, all within a single execution cycle, it is possible for the data ALU to perform a multiplication operation, for the AGU to generate up to two addresses, and for the program controller to prefetch the next instruction.

The major components of the 56800E core include the following:

- Address buses
- Data buses
- Data Arithmetic Logic Unit (ALU)

- Address Generation Unit (AGU)
- Program controller and hardware looping unit
- Bit-manipulation unit
- Enhanced OnCE debugging module
- Clock generation
- Reset circuitry

1.4.5 Address Buses

The core contains three address buses:

1. Program memory Address Bus (PAB)
2. Primary Data Address Bus (XAB1)
3. Secondary Data Address Bus (XAB2)

The PAB is 21 bits wide. It is used to address (16-bit) words in program memory. The two 24-bit data address buses permit two simultaneous accesses to data (X) memory. The XAB1 bus can address byte, word, and long data types. The XAB2 bus is limited to (16-bit) word accesses.

All three buses address on- and off-chip memory on devices containing an external bus interface unit. The 56857 device does not provide external addressing. The XAB2 can not go off-chip.

1.4.6 Data Buses

Data transfers inside the chip occur over the following buses:

- Two unidirectional 32-bit buses:
  - Core Data Bus for Reads (CDBR)
  - Core Data Bus for Writes (CDBW)
- Two unidirectional 16-bit buses:
  - Secondary X Data Bus (XDB2)
  - Program Data Bus (PDB)
- IPBus interface

Data transfers between the data ALU and data memory use the CDBR and CDBW when a single memory read or write is performed. When two simultaneous memory reads are performed, the transfers use the CDBR and XDB2 buses. All other data transfers to core blocks occur using the CDBR and CDBW buses. Peripheral transfers occur through the IPBus interface. Instruction word fetches occur over the PDB.

This bus structure supports up to three simultaneous 16-bit transfers. Any one of the following can occur in a single clock cycle:

- One instruction fetch
- One read from data memory
- One write to data memory
- Two reads from data memory
- One instruction fetch and one read from data memory
- One instruction fetch and one write to data memory
- One instruction fetch and two reads from data memory

An instruction fetch will take place on every clock cycle, although it is possible for data memory accesses to be performed without an instruction fetch. Such accesses typically occur when a hardware loop is executed and the repeated instruction is only fetched on the first loop iteration.

1.4.7 Data Arithmetic Logic Unit (Data ALU)

The data Arithmetic Logic Unit (ALU) performs all of the arithmetic, logical, and shifting operations on data operands. The data ALU contains the following components:

- Three, 16-bit data registers (X0, Y0, and Y1)
- Four, 36-bit accumulator registers (A, B, C, and D)
- One Multiply-Accumulator (MAC) unit
- A single-bit accumulator shifter
- One arithmetic and logical multi-bit shifter
- One MAC output limiter
- One data limiter

All in a single instruction cycle, the data ALU can perform multiplication, multiply-accumulation, with positive or negative accumulation, addition, subtraction, shifting, and logical operations. Division and normalization operations are provided by iteration instructions. Signed and unsigned multi-precision arithmetic is also supported. All operations are performed using two’s-complement fractional or integer arithmetic.

Data ALU source operands can be 8, 16, 32, or 36 bits in size and can be located in memory, in immediate instruction data, or in the data ALU registers. Arithmetic operations and shifts can have 16-, 32-, or 36-bit results. Logical operations are performed on 16- or 32-bit operands and yield results of the same size. The results of data ALU operations are stored either in one of the data ALU registers or directly in memory.

1.4.8 Address Generation Unit (AGU)

The Address Generation Unit (AGU) performs all of the calculations of effective addresses for data operands in memory. It contains two address ALUs, allowing up to two 24-bit addresses to be generated every instruction cycle:

1. One for either the Primary Data Address Bus (XAB1), or the Program Address Bus (PAB)
2. One for the Secondary Data Address Bus (XAB2)

The address ALU can perform both linear and modulo address arithmetic. The AGU operates independently of the other core units, minimizing address-calculation overhead.

The AGU can directly address 2^{24} (16M) words on the XAB1 and XAB2 buses. It can access 2^{21} (2M) words on the PAB. The XAB1 bus can address byte, word, and long data operands. The PAB and XAB2 buses can only address words in memory.

The AGU consists of the following registers and functional units:

- Seven, 24-bit address registers (R0–R5 and N)
- Four, 24-bit shadow registers for address registers (for R0, R1, M, and M01
- A 24-bit dedicated Stack Pointer (SP) register
- Two offset registers (N and N3)
- A 16-bit modifier register (M01)
- A 24-bit adder unit
- A 24-bit modulo arithmetic unit

Each of the address registers (R0–R5) can contain either data or an address. All of these registers can provide an address for the XAB1 and PAB address buses; addresses on the XAB2 bus are provided by the R3 register. The N offset register can be used either as a general-purpose address register or as an offset or update value for the addressing modes supporting those values. The second 16-bit offset register (N3) is used only for offset or update values. The modifier register (M01) selects between linear and modulo address arithmetic.

1.4.9 Program Controller and Hardware Looping Unit

The Program Controller is responsible for instruction fetching and decoding, interrupt processing, hardware interlocking, and hardware looping. Actual instruction execution takes place in the other core units, such as in the data ALU, AGU, or bit-manipulation unit.

The Program Controller contains the following:

- An instruction latch and decoder
- The hardware looping control unit

- Interrupt control logic
- A Program Counter (PC)
- Two special registers for Fast Interrupts:
  - Fast Interrupt Return Address Register (FIRA)
  - Fast Interrupt Status Register (FISR)
- Seven user-accessible status and control registers:
  - two-level deep Hardware Stack (HWS)
  - Loop Address (LA) register
  - Loop Address (LA2) register 2
  - Loop Count (LC) register
  - Loop Count (LC2) register 2
  - Status Register (SR)
  - Operating Mode Register (OMR)

The Operating Mode Register (OMR) is a programmable register to control the operation of the 56800E core, including the memory-map configuration. The initial operating mode is typically latched on reset from an external source; it can subsequently be altered under program control.

The Loop Address (LA) and Loop Count (LC) registers work in conjunction with the hardware stack to support no-overhead hardware looping. The Hardware Stack is an internal Last-In-First-Out (LIFO) buffer consisting of two, 24-bit words and stores the address of the first instruction of a hardware DO loop. When executing the DO instruction begins a new Hardware Loop, the address of the first instruction in the Loop is pushed onto the Hardware Stack. When a Loop finishes normally or an ENDDO instruction is encountered, the value is popped from the Hardware Stack. This process allows one Hardware DO Loop to be nested inside another.

1.4.10 Bit Manipulation Unit

The bit-manipulation unit performs bit field operations on data memory words, peripheral registers, and registers within the 56800E core. It is capable of testing, setting, clearing, or inverting individual or multiple bits within a 16-bit word. The bit-manipulation unit can also test bytes for branch-on-bit field instructions.

1.4.11 Enhanced On-Chip Emulation (EOnCE) Module

The Enhanced On-Chip Emulation (EOnCE) module allows user interaction in a debug environment with the 56800E core and its peripherals. Its capabilities include:

- Examining registers
- Memory, or on-chip peripherals
- Setting breakpoints in memory
- Stepping or tracing instructions

It provides simple, inexpensive, and speed independent access to the 56800E core for sophisticated debugging and economical system development. The JTAG port allows access to the Enhanced OnCE module and through the 5685x device to its target system, retaining debug control without sacrificing other user accessible on-chip resources. This technique eliminates the costly cabling and the access to processor pins required by traditional emulator systems. The Enhanced OnCE interface is fully described in the *DSP56800E Reference Manual* (DSP56800ERM).

1.4.12 Clocks

1.4.12.1 On-Chip Clock Synthesis Block

The clock synthesis module generates the clocking for the 5685x family of devices. It generates the master clock used by the System Integration Module (SIM) to derive the system and peripheral clocks. It also generates the time of day clock used by time-based modules like TOD and COP. It contains an oscillator module to apply the clock. It also contains a PLL with the ability to multiply-up the frequency. The PLL can also be bypassed and scaled to lower power consumption on the 5685x device. The CGM module selects which clock is routed to the master clock output. It also selects and configures the time of day clock prescaler and PLL.

1.4.12.2 Oscillator

The 5685x device is clocked either from an external crystal or external clock input:

- Crystal oscillator uses a 2 - 4MHz crystal
- Ceramic resonator can be used in place of the crystal
- There are separate power and ground for the oscillator and PLL
- Oscillator input can be directly clocked at up to 240MHz

1.4.12.3 PLL

The PLL in the 56800E core provides the following features:

- The PLL generates output frequencies up to 240MHz from a 2 - 4MHz input
- The PLL can be bypassed to use oscillator or prescalar outputs directly

1.4.12.4 Clock Generation Module

This contains registers used to control clock generation and select clock sources:

- There is a choice of time of day clock prescalers
- The PLL frequency postscaler and enable control
- Selection of master clock source to be PLL ,or OSC and *glitch-free* clock switching
- PLL postscaler supports PLL output division by 1, 2, 4, 8, 16, 64,or 128

1.4.12.5 Time of Day Clock Options

These options provide:

- A choice of two TOD clock prescalers
- Low power/128 prescaler in OSC module for input clock frequencies to 4MHz
- High frequency two stage/1-4096 and /2 prescaler in CGM module for input clock frequencies to 240MHz OSC TOD prescaler

1.4.13 Resets

The 5685x device reset circuitry provides these features:

- Integrated POR release occurs when V_{DD} exceeds 1.35V and V_{DDA} exceeds 2.45V
- Reset pin
- Software reset
- COP reset

1.4.14 IPBus Bridge

The IPBus architecture supports a variety of on-chip peripherals, including:

- External Memory Interface (EMI) module
- Serial Communication Interface (SCI) module
- 16-bit Timer (TMR) module
- Computer Operating Properly (COP) module
- Time of Day (TOD) module

- Enhanced Synchronous Serial Interface (ESSI) module
- Serial Peripheral Interface (SPI) module
- Programmable General-Purpose I/O (GPIO) module
- Eight-bit Parallel Host Interface (HI8)

1.5 System Bus Controller

The System Bus Controller (SBC) controls a number of functions essential to the transfer of data between the core, DMA controllers and memory within the 56853/854/855/857/858 systems.

1.5.1 Operation

The SBC performs a number of central roles in the transfer of data between either the core or DMA controller and memory space. In every clock cycle, the SBC determines whether the core or DMA is bus master, which memory device in any space is active, and if the core clock is active. Through these actions, all data transfers are completed at maximum bus efficiency.

The SBC is capable of supporting one core and one DMA controller of up to six channels (operating in X1 data space only):

- Up to three program memory address spaces
- Up to two X1 memory address spaces
- Single X2 memory address space

Either the core or DMA controller are capable of initiating memory transfers. Bus mastership is arbitrated by the SBC, and the device which is permitted to access the bus on any given cycle is referred to as the active bus master.

1.5.2 IPBus Bridge (IPBB)

The IPBus Bridge (IPPB) provides a means for communication between the high speed core and the low-bandwidth devices on the IP peripheral bus. Among other functions, the bridge is responsible for maintaining an orderly and synchronized communication between devices on both sides running at two different clock frequencies.

Figure 1-8 denotes the position and interface of the IPBus Bridge with other main blocks within the chip.

Other connections in the figure not pertaining to the primary function of the bridge are omitted for clarity; nevertheless, they will be discussed as appropriate. A brief description of bridge’s interface with various main components on both sides is also provided.

1.5.2.1 System Side Operation

On the system side, the IPBus Bridge operates at core frequency and fully supports pipelined communication with the core. The bridge acts as a slave device on this bus. The bridge is responsible for initiating IPBus transactions only per requests initiated by the core, or other system bus masters.

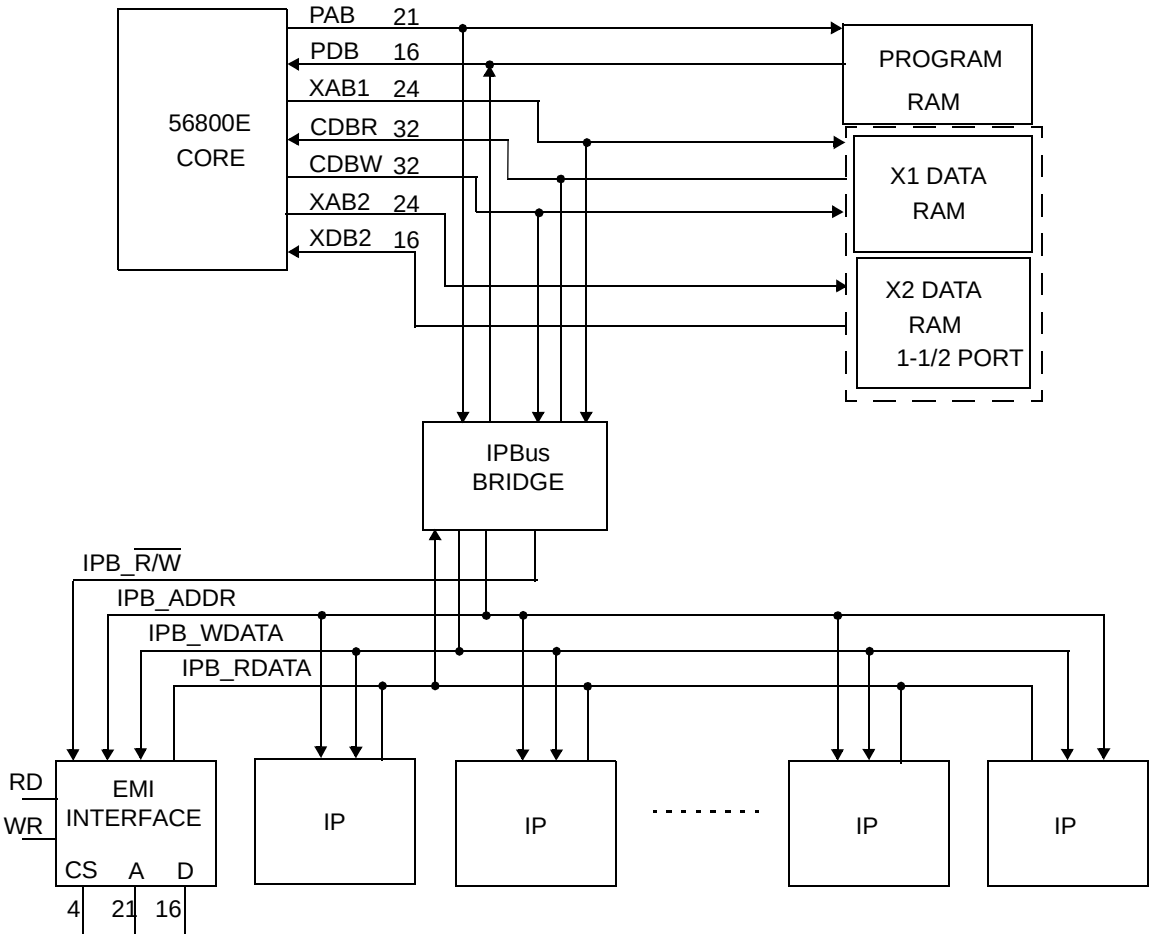


Figure 1-8. IPBus Bridge Interface With Other Main Components System Side Operation

1.5.2.2 Peripheral Side Operation

On the peripheral side, the IPBus Bridge accesses various devices through a standard non-pipelined IPBus interface. Separate bus lines are used for read and write transactions. The IPBus Bridge also interfaces with an External Memory Interface (EMI) block. The IPBus operates at half of the core frequency.

1.6 5685x Memory

These lists provide features of the 56853/854/55/857/858's memory modules:

- 56853
  - 12K × 16-bit Program SRAM
  - 4K × 16-bit Data SRAM
  - 1K × 16-bit boot ROM
- 56854
  - 16K × 16-bit Program SRAM
  - 16K × 16-bit Data SRAM
  - 1K × 16-bit boot ROM
- 56855
  - 24K × 16-bit Program SRAM
  - 24K × 16-bit words of Data SRAM
  - 1K × 16-bit boot ROM
- 56857
  - 40K × 16-bit Program SRAM
  - 24K × 16-bit Data SRAM
  - 1K × 16-bit boot ROM
- 56858
  - 40K x 16-bit Program SRAM
  - 24K x 16-bit Data SRAM
  - 1K x 16-bit boot ROM

1.6.1 Program SRAM

- Single Port RAM is compatible with the pipelined program bus structure
- Single cycle reads at 120MHz

1.6.2 Data SRAM

- Single read, dual read or single write memory compatible with the pipelined data bus structure
- Single cycle reads/writes at 120MHz

1.6.3 Boot ROM

- Single port ROM is compatible with the pipelined program bus structure
- Single cycle reads at 120MHz

1.7 56853 Peripheral Blocks

The 56853 provides these peripheral blocks:

- Two Serial Communication Interfaces (SCI0 and SCI2), each with two pins, or four additional GPIO lines
- Enhanced Synchronous Serial Interface (ESSI0), with six pins or additional GPIO lines
- General Purpose 16-bit Quad Timer (TMR), with four pins or additional GPIO lines
- One Serial Peripheral Interface (SPI), with four pins or additional GPIO lines
- Eight-bit Host Interface (HI8) with 16 pins, or 16 additional GPIO lines
- Interrupt Controller
- Computer Operating Properly (COP)/Watchdog Timer
- Time of Day (TOD)
- Clock Generator
- System Integration Module (SIM)
- External Memory Interface (EMI)
- JTAG/Enhanced On-Chip Emulation (EOnCE) for unobtrusive, real-time debugging

1.8 56854 Peripheral Blocks

The 56854 provides these peripheral blocks:

- Two Serial Communication Interfaces (SCI), each with two pins, or four additional GPIO lines
- Enhanced Synchronous Serial Interface (ESSI0), with six pins or additional GPIO lines
- General Purpose 16-bit Quad Timer (TMR) with four pins or additional GPIO lines
- Serial Port Interface (SPI) with four pins or additional GPIO lines
- Eight-bit Parallel Host Interface (HI8) with 16 pins, or 16 additional GPIO lines
- Interrupt Controller
- Computer Operating Properly (COP)/Watchdog Timer
- Time of Day (TOD)
- Clock Generator
- System Integration Module (SIM)

- External Memory Interface (EMI)
- JTAG/Enhanced On-Chip Emulation (EOnCE) for unobtrusive, real-time debugging

1.9 56855 Peripheral Blocks

The 56855 provides these peripheral blocks:

- Two Serial Communication Interfaces (SCI0 and SCI2), each with two pins, or four additional GPIO lines
- Enhanced Synchronous Serial Interface (ESSI0), with six pins or additional GPIO lines
- General Purpose 16-bit Quad Timer with one pin or additional GPIO line
- Interrupt Controller
- Computer Operating Properly (COP)/Watchdog Timer
- Time of Day (TOD)
- Clock Generator
- System Integration Module (SIM)
- External Memory Interface (EMI)
- JTAG/Enhanced On-Chip Emulation (EOnCE) for unobtrusive, real-time debugging

1.10 56857 Peripheral Blocks

The 56857 provides these peripheral blocks:

- Two Serial Communication Interfaces (SCI), each with two pins, or four additional GPIO lines
- Two Enhanced Synchronous Serial Interfaces (ESSI0 and ESSI1), each with six pins, or 12 additional GPIO lines
- General Purpose 16-bit Quad Timer (TMR) with four pins or additional GPIO lines
- Serial Port Interface (SPI) with four pins or additional GPIO lines
- Eight-bit Parallel Host Interface (HI8) with 16 pins, or 16 additional GPIO lines
- Interrupt Controller
- Computer Operating Properly (COP)/Watchdog Timer
- Time of Day (TOD)
- Clock Generator
- System Integration Module (SIM)
- External Memory Interface (EMI)
- JTAG/Enhanced On-Chip Emulation (EOnCE) for unobtrusive, real-time debugging

1.11 56858 Peripheral Blocks

The 56858 provides these peripheral blocks:

- Two Serial Communication Interfaces (SCI), each with two pins, or four additional GPIO lines
- Two Enhanced Synchronous Serial Interfaces (ESSI0 and ESSI1), each with six pins, or 12 additional GPIO lines
- General Purpose 16-bit Quad Timer (TMR) with four pins or additional GPIO lines
- Serial Port Interface (SPI) with four pins or additional GPIO lines
- Eight-bit Parallel Host Interface (HI8) with 16 pins, or 16 additional GPIO lines
- Interrupt Controller
- Computer Operating Properly (COP)/Watchdog Timer
- Time of Day (TOD)
- Clock Generator
- System Integration Module (SIM)
- External Memory Interface (EMI)
- JTAG/Enhanced On-Chip Emulation (EOnCE) for unobtrusive, real-time debugging

1.12 Peripheral Descriptions

The IPBus bridge converts program and data memory accesses to the IPBus-compliant interface for peripherals and external EMI. This IPBus bridge allows for communication between the core and peripherals utilizing the CDBR for data and XAB for addresses. To access program space through the EMI, use PAB and PDB buses. Peripherals run off the IPBus clock at up to 60MHz. The IPBus clock frequency is half of the system clock frequency.

1.12.1 External Memory Interface

The EMI design includes these distinctive features:

- Programmable wait states for slower memories (up to 16 IP_CLK access time)
- Contains four programmable chip selects
- No external glue logic required for typical systems, if the chip selects are used
- Chip selects may be independently programmed with various features
- Program or data space selection
- Programmable byte enables, to support 8-bit wide external memories

1.12.2 General Purpose Input/Output Port (GPIO)

- 56853
  - 41 shared GPIO, multiplexed with other peripherals
  - Each bit may be individually configured as an input or output
  - Selectable enable for pull-up resistors
- 56854
  - 41 shared GPIO, multiplexed with other peripherals
  - Each bit may be individually configured as an input or output
  - Selectable enable for pull-up resistors
- 56855
  - 18 shared GPIO, multiplexed with other peripherals
  - Each bit may be individually configured as an input or output
  - Selectable enable for pull-up resistors
- 56857
  - 47 GPIO, multiplexed with other peripherals (43 multiplexed and 4 dedicated)
  - Each bit may be individually configured as an input or output
  - Selectable enable for pull-up resistors
- 56858
  - 47 GPIO, multiplexed with other peripherals
  - Each bit may be individually configured as an input or output
  - Selectable enable for pull-up resistors

1.12.3 Serial Communications Interface (SCI)

Each of the chips in the 56800E family has two Serial Communication Interfaces (SCI0 and SCI1). SCI features include:

- Asynchronous operation
- Baud rate generation
- IR interface support

1.12.4 Enhanced Synchronous Serial Interface (ESSI)

The ESSI is a full-duplex, serial port designed to allow digital signal controllers (DSCs) to communicate with a variety of serial devices, including industry-standard codecs, other DSCs, and microprocessors. It is typically used to transfer samples in a periodic manner. The ESSI consists of independent transmitter and receiver sections with independent clock generation and frame synchronization. ESSI features include:

- Independent (asynchronous) or shared (synchronous) transmit and receive sections with separate or shared internal/external clocks and frame syncs
- Normal mode operation using frame sync
- Network mode operation allowing multiple devices to share the port with as many as 32-time slots
- Network mode enhancements
  - time slot mask registers (receive and transmit)
  - end of frame interrupt
- Programmable internal clock divider
- Programmable word length (8, 10, 12, or 16 bits)
- Program options for frame sync and clock generation
- ESSI power-down feature
- Audio enhancements
  - three transmitters per ESSI (for six-channel surround sound)

1.12.5 Quad Timer (TMR) Module

Each chip in the 56800E family contains a general purpose 16-bit TMR module.

The Quad Timer (TMR) module has four external signals capable of being used as either inputs or outputs. They may also be used to interface to the Peripheral Bus. The 855 has only one external signal. TMR features include:

- Four, 16-bit counters/timers
- Count up/down
- Counters are cascadable
- Programmable count modulo
- Max count rate equals peripheral clock/2 for external clocks
- Max count rate equals peripheral clock for internal clocks
- Count once or repeatedly

- Counters are preloadable
- Counters can share available input pins
- Separate prescaler for each counter
- Each counter has capture and compare capability

1.12.6 Serial Peripheral Interface (SPI)

The Serial Peripheral Interface (SPI) is an independent serial communications subsystem allowing full-duplex, synchronous, serial communication between the signal controller and peripheral devices, including other DSCs. Software can poll SPI status flags or SPI operation can be interrupt driven. This block contains four, 16-bit memory mapped registers for control parameters, status, and data transfer. SPI features include:

- Full-duplex operation
- Master and slave modes
- Double-buffered operation with separate transmit and receive registers
- Programmable length transmissions (2 to 16 bits)
- Programmable transmit and receive shift order (MSB first or last bit transmitted)
- Eight master mode frequencies (maximum = bus frequency/2^1)
- Maximum slave mode frequency = bus frequency
- Clock ground for reduced Radio Frequency (RF) interference
- Serial clock with programmable polarity and phase
- Two separately enabled interrupts
  - SPRF (SPI Receiver Full)
  - SPTE (SPI Transmitter Empty)
- Mode fault error flag interrupt capability
- Direct Memory Access (DMA) capability for both the transmitter and receiver
- Wired OR mode functionality to enabling connection to multiple SPIs

1. This frequency may be further limited by the GPIO function.

1.12.7 Host Interface 8 (HI8)

Host Interface 8 (HI8) is an 8-bit, bidirectional data bus used to transfer data between the host processor and the DSC. To accomplish this, a series of registers are provided to establish the control desired, to monitor the status of the data transfer and to perform the data transfer itself. Registers are divided into two groups:

1. Host side
2. DSC side

Both groups allow the host port to operate asynchronously to core clock. Generally, the mode of operation is set using the host side control. The DSC side control provides the selectivity for masking or unmasking the various DSC interrupt sources. There are various functional modes possible in the transfer of data: 8-bit vs. 16-bit data transfers, DMA vs. single strobe or dual strobe transactions, and interrupt versus polling status interrogation.

1.12.8 COP/Watchdog Timer Module

The Computer Operating Properly (COP) module monitors processor activity and provides an automatic reset signal if a failure occurs. Please reference **Chapter 16, Reset, Low Voltage, Stop and Wait Operations.**

- 16-bit counter to provide 65536 different time-out periods
- Programmable wait and stop mode operation
- Programmable time-out period from 32µs to 2.1 sec. with a resolution of 32µs

1.12.9 Time of Day

- Sequence counters to track seconds, minutes, hours, and days
- Programmable prescaler to generate 1Hz clock
- Generates interrupts capable of pulling the part out of Wait or Stop modes
- Capability to track time up to 179.5 years
- Configurable alarm and one second interrupts

1.12.10 System Integration Module

The System Integration Module (SIM) is responsible for several system control functions including:

- Clock generation
- Reset generation
- Power mode control

- Boot mode control
- Memory map control
- External I/O configuration

SIM features include:

- Four system bus clocks with pipeline hold off support at master clock frequency/2
- Three system clocks for non pipelined interfaces at master clock frequency/2
- A peripheral bus (IPBus) clock, both on standard and inverted versions at master clock frequency/4
- An external clock output with disable at master clock frequency/5
- A core stall control used to stall the 56800E core system clock for DMA mastership
- Three power modes to control power utilization
- Controls to enable/disable the 56800E core Wait and Stop instructions
- Software initiated reset
- Controls to redirect internal data and/or program RAM accesses to the external memory interface
- Software Boot mode Control Register, initialized at any reset except COP reset from external pads via SIM inputs MODEA, B, and C
- A hold off output to coordinate system and peripheral buses
- Two 16-bit registers reset only by a power-on reset usable for general purpose software control

1.12.11 JTAG/Enhanced OnCE Port

The JTAG/Enhanced OnCE port allows insertion of the 5685x devices into a target system while retaining debug control. The JTAG port provides board-level testing capability for scan-based emulation compatible with the IEEE 1149.1a-1993 IEEE Standard Test Access Port and Boundary Scan Architecture specification defined by the JTAG. Five dedicated pins interface to a TAP containing a 16-state controller.

The EOnCE module allows the user to interact in a debug environment with the 56800E core and its peripherals nonintrusively. Its capabilities include:

- Examining registers, memory, or on-chip peripherals
- Setting breakpoints in memory
- Stepping or tracing instructions

It provides simple, inexpensive, and speed-independent access to the 56800E core for sophisticated debugging and economical system development. The JTAG/EOnCE port provides access to the EOnCE module. The JTAG/EOnCE port retains debug control without sacrificing other accessible on-chip resources through the 5685x devices to its target system.

1.12.12 Six-Channel DMA Controller

- The DMA operates independently of the CPU.
- The 5685x DMA includes six channels. The DMA can handle the contexts of six independent block transfers.
- The DMA Controller has access to 33 percent of the system bus traffic. This one-quarter duty cycle is governed by the 5685x System Bus Controllers.
- Each channel has independently programmable peripheral selection.
- Each channel's source and destination address registers has configurable indices.
- For each memory *fetch* and *put*, the address may remain constant, be post incremented, or post decremented.
- Each read or write transfer may be initiated by selected events: specified peripheral requests or direct CPU triggering may launch a new DMA transaction.
  - Upon completion of a block transfer, each DMA channel may send an interrupt to the CPU.
  - The DMA includes a *circular queue* operational mode providing continuous DMA operation with no additional processor intervention.

1.12.13 Peripheral Interrupts/Interrupt Controller

The peripherals on the 5685x use the ITCN module to interface to the interrupt signal found on the 56800E core. Each peripheral has its own interrupt vector, often more than one interrupt vector for each peripheral, and can selectively be enabled or disabled via the IPR registers. The Interrupt Controller (ITCN) module design includes these distinctive features:

- Programmable priority levels for each IRQ
- Two programmable Fast Interrupts
- Notification to SIM module to restart clocks out of Wait and Stop modes

1.13 56800E Programming Model

The programming model for the registers in the 56800E core is shown in **Figure 1-9.**

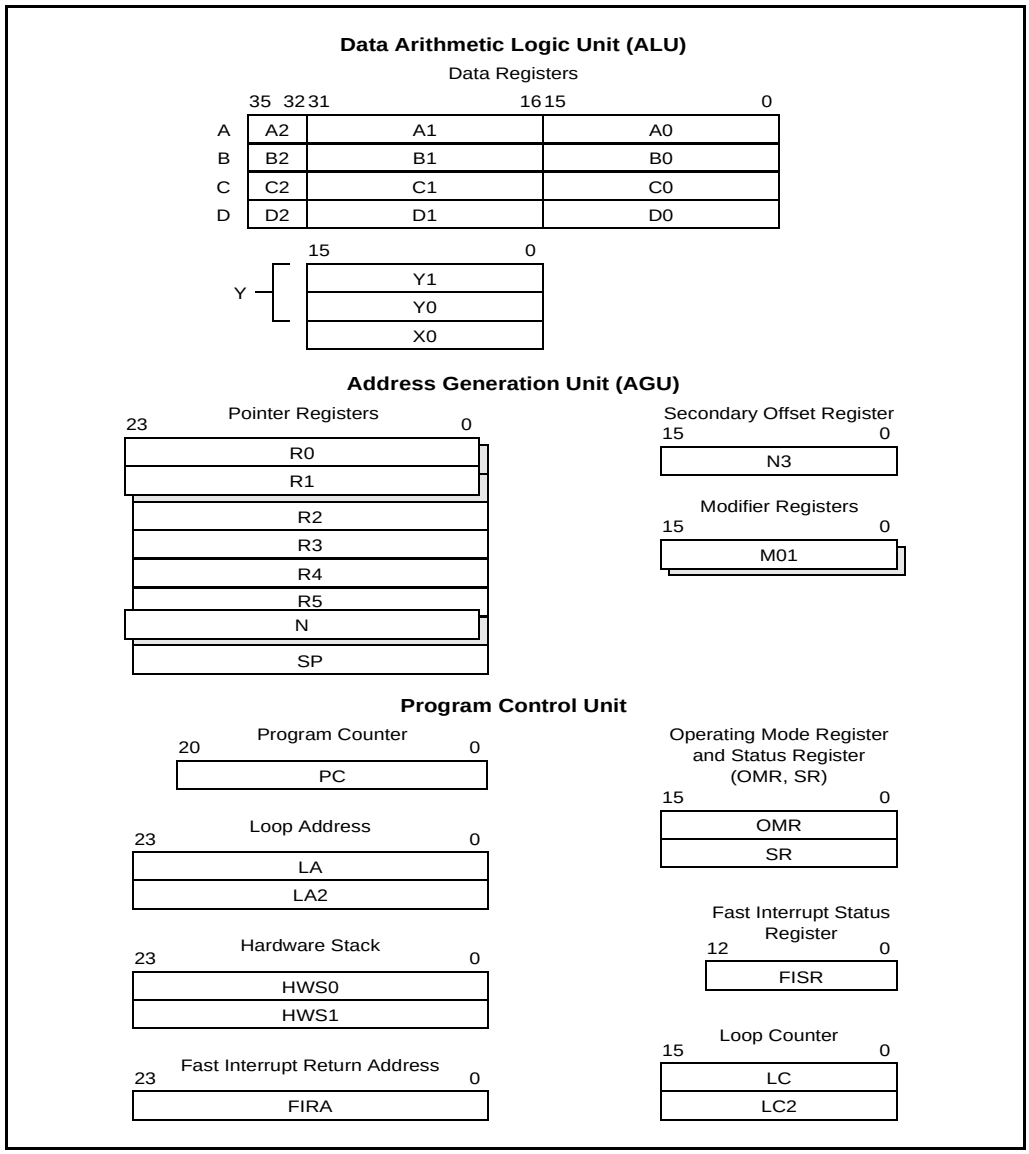


Figure 1-9. Register Programming Model for the 5685x

Chapter 2
Pin Descriptions

2.1 Introduction

This chapter details the input and output signals and functions of 568x packaging. Depending on the device, their pins allow a variety of functions and capabilities. Each is detailed in this chapter.

In addition to the available peripherals, up to 47-general purpose pins are also available when not required for other purposes. The remaining signal pins are dedicated to one function. The input and output signals of these packages are organized into functional groups, shown in **Table 2-1** and illustrated in **Figure 2-1** through **Figure 2-5.** Each table row describes the package pins and the signal or signals present in **Table 2-2** through **Table 2-15.** Interface signals have these general characteristics:

- Pins listed here are pulled high with an on-chip resistor
  - TDI
  - TMS
  - $\overline{\text{TRST}}$
  - $\overline{\text{DE}}$
- This pin is pulled low with an on-chip resistor:
  - TCK
- These pins are pulled high by the device during hardware reset:
  - $\overline{\text{RD}}$
  - $\overline{\text{WR}}$
- These pins have the internal pull-ups permanently disabled:
  - MODA
  - MODB
  - MODC
- General Purpose I/O pins have programmable pull-up resistors

Table 2-1. Functional Group Pin Allocations

Functional Group	Number of Pins					Detailed Description
	853	854	855	857	858	
Power (V_{DD}, V_{DDA}, or V_{DD} Core)	18	18	15	21	21	Table 2-2
Ground (V_{SS} or V_{SSA})	17	17	15	19	24	Table 2-3
External Chip Select Signals*	39	39	39	—	39	Table 2-4
External Bus Control Signals	4	4	4	4	4	Table 2-5
Host Interface (HI)*	16	16	—	16	16	Table 2-6
Quad Timer Module (TMR) Port*	4	4	1	4	4	Table 2-7
Interrupt and Program Control	7	7	7	7	7	Table 2-8
Serial Communication Interface (SCI0) Ports*	2	2	2	2	2	Table 2-9
Serial Communication Interface (SCI1) Ports*	2	2	2	2	2	Table 2-10
Enhanced Synchronous Serial Interface (ESSI0) Port*	6	6	6	6	6	Table 2-11
Enhanced Synchronous Serial Interface (ESSI1) Port*	—	—	—	6	6	Table 2-12
Serial Peripheral Interface (SPI) Port*	4	4	—	4	4	Table 2-13
Clock and Phase Lock Loop (PLL)	3	3	3	3	3	Table 2-14
JTAG/EOnCE	6	6	6	6	6	Table 2-15

*Alternately, General-Purpose I/O pins

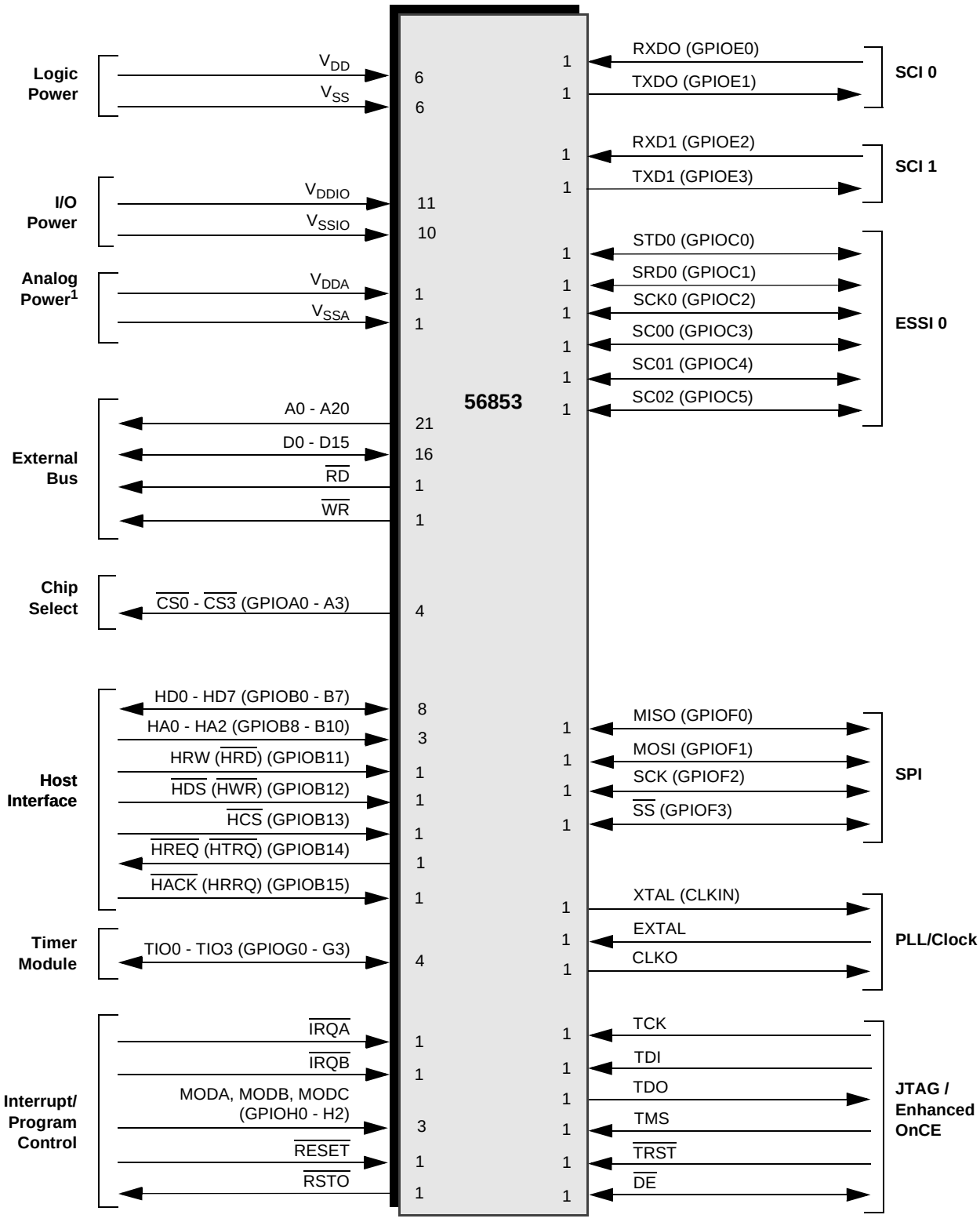


Figure 2-1. 56853 Signals Identified by Functional Group[2]

1. Specifically for PLL, OSC, and POR.
2. Alternate pin functions are shown in parentheses.

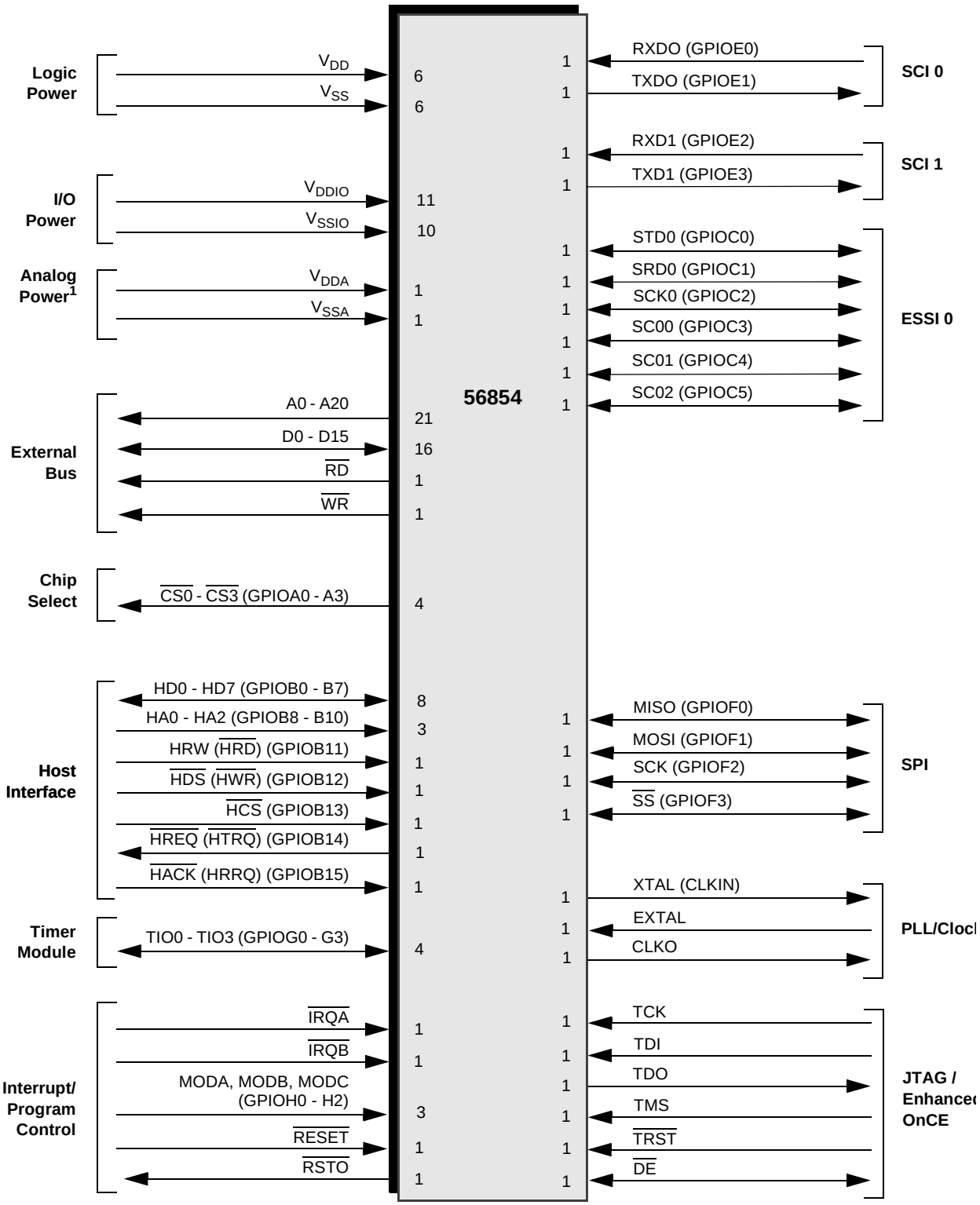


Figure 2-2. 56854 Signals Identified by Functional Group[2]

1. Specifically for PLL, OSC, and POR.
2. Alternate pin functions are shown in parentheses.

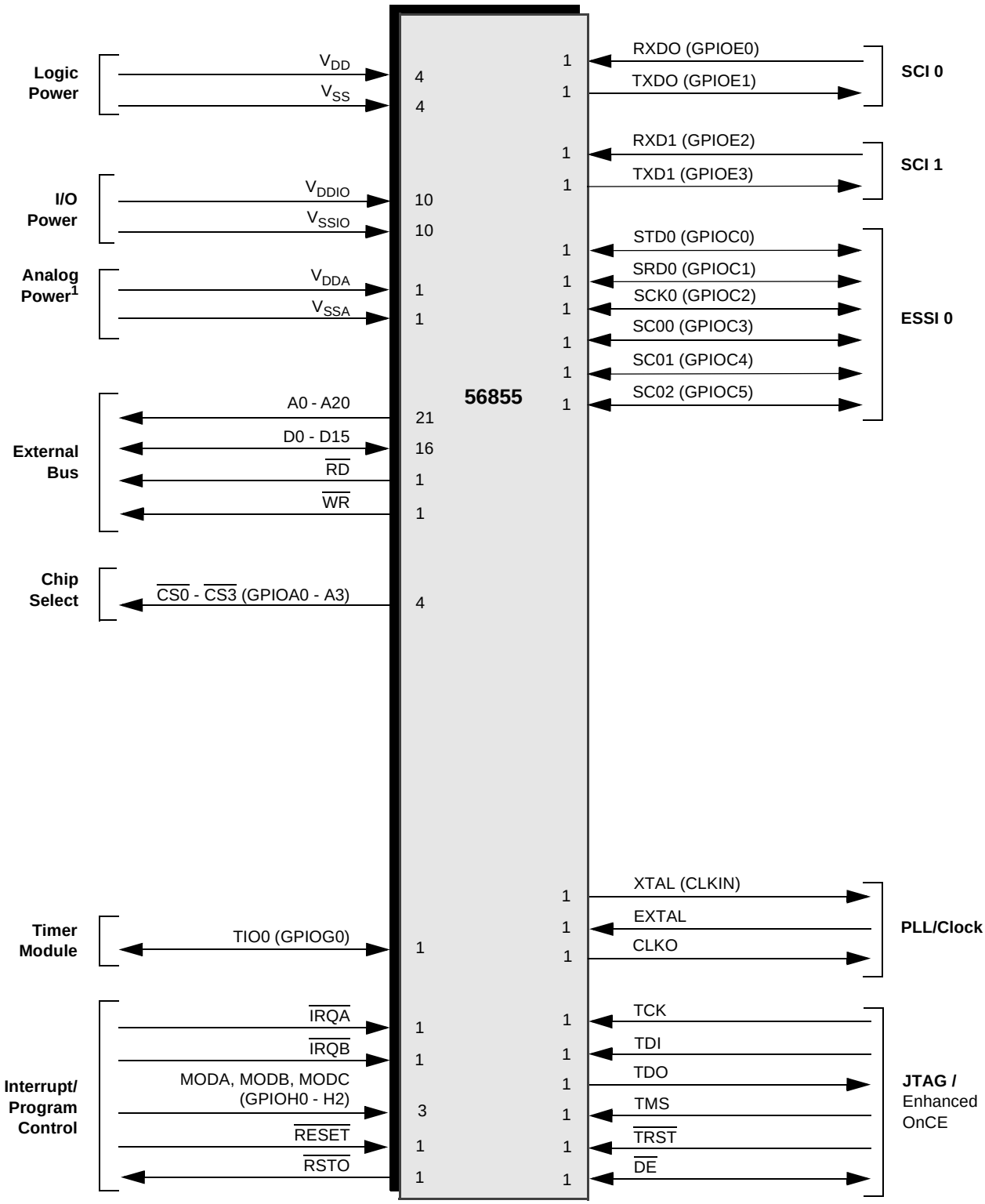


Figure 2-3. 56855 Signals Identified by Functional Group[2]

1. Specifically for PLL, OSC, and POR.
2. Alternate pin functions are shown in parentheses.

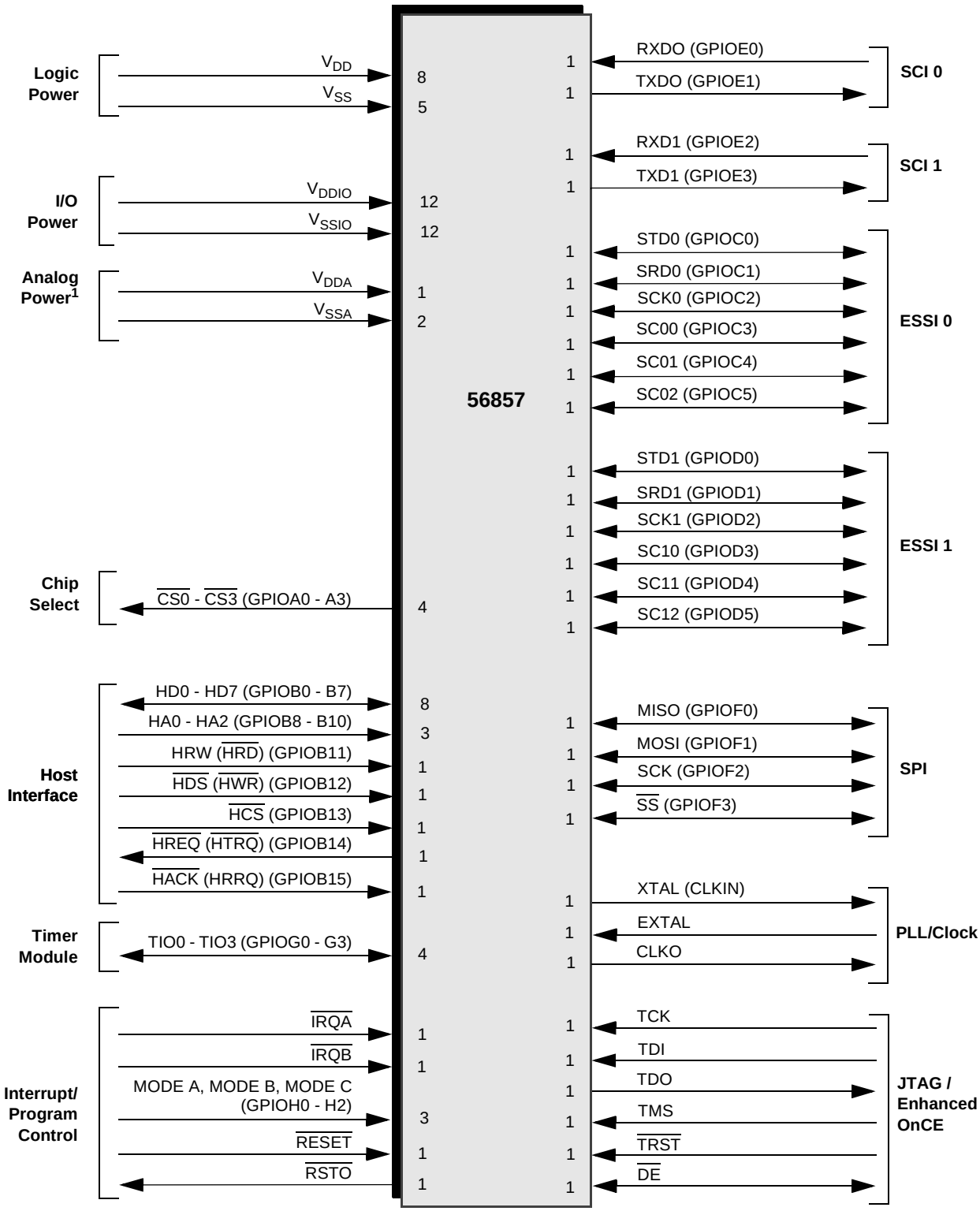


Figure 2-4. 56857 Signals Identified by Functional Group[2]

1. Specifically for PLL, OSC, and POR.
2. Alternative pin functions are shown in parentheses.

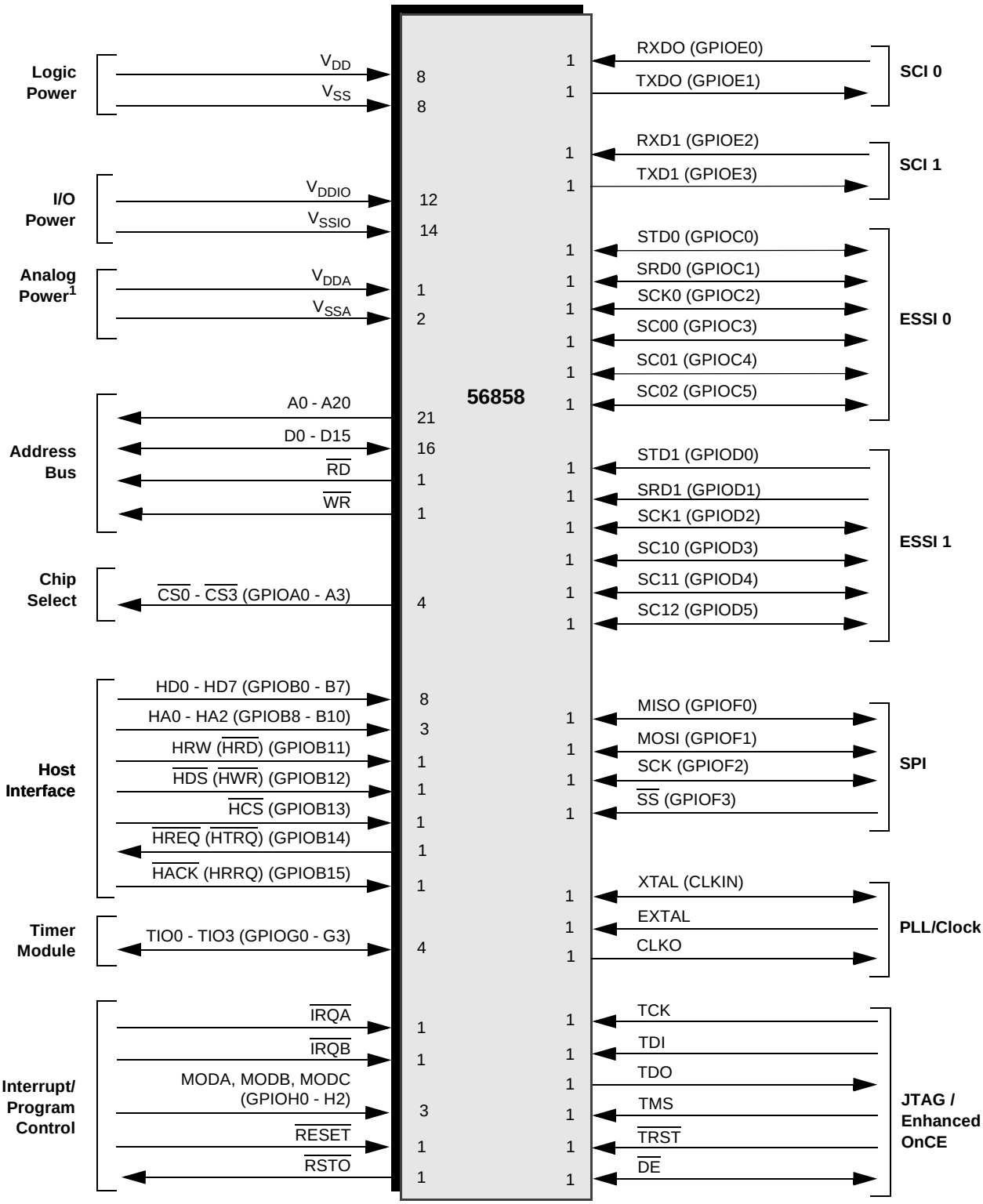


Figure 2-5. 56858 Signals Identified by Functional Group[2]

1. Specifically for PLL, OSC, and POR.
2. Alternate pin functions are shown in parentheses.

2.2 Signal and Package Information

All digital inputs have a weak internal pull-up circuit associated with them. These pull-up circuits are enabled by default. Exceptions:

1. When a pin has GPIO functionality, the pull-up may be disabled under software control
2. Mode pins D13, D14, and D15 have no pull-up
3. TCK has a weak pull-down circuit always active
4. Bidirectional I/O pull ups automatically disabled when the output is enabled

The following tables are presented consistently with the *Signals Identified by Functional Group* figure.

1. **BOLD** entries in the Type column represents the state of the pin just out of reset.
2. Output(Z) means an output is in a High-Z condition.

2.3 Power, Ground and Peripheral Signals

The following tables illustrate power, ground, and bypass capacitor pinout data. **Table 2-2** through **Table 2-9** displays the physical layout of various grounds and voltage input signals associated with the input/output ring. Signal names in parentheses on the same table row provide the GPIO alternative pin function.

Table 2-2. Power Inputs

Signal Name	Signal Description	853 Pins	854 Pins	855 Pins	857 Pins	858 Pins
V_{DD}	**Power—**These pins provide power to the internal structures of the chip. They all should be attached to V_{DD}.	6	6	4	8	8
V_{DDIO}	**Power—**These pins provide power for all I/O and ESD structures of the chip. They all should be attached to V_{DDIO} (3.3V).	11	11	10	12	12
V_{DDA}	**Analog Power**—These pins supply analog power.	1	1	1	1	1

Table 2-3. Grounds

Signal Name	Signal Description	853 Pins	854 Pins	855 Pins	857 Pins	858 Pins
V_{SS}	**Ground—**These pins provide grounding for the internal structures of the chip and should all be attached to V_{SS}.	6	6	4	5	8
V_{SSIO}	**Ground—**These pins provide grounding for all I/O and ESD structures of the chip. They all should be attached to V_{SSIO}.	10	10	10	12	14
V_{SSA}	**Analog Power**—These pins supply an analog ground.	1	1	1	2	2

Table 2-4. External Bus Control Signals

Signal Name	Signal Type	Signal Description	853 Pins	854 Pins	855 Pins	857 Pins	858 Pins
A0-A20	Output(Z)	**Address Bus—**These signals specify a word address for external program or data memory access.	21	21	21	—	21
D0-D15	**Input** / Output(Z)	**Data Bus—**These pins provide the bidirectional data for external program or data memory accesses.	16	16	16	—	16
$\overline{RD}$	Output	**Read Enable—**is asserted during external memory read cycles. This signal is pulled high during reset.	1	1	1	—	1
$\overline{WR}$	Output	**Write Enable—**is asserted during external memory write cycles. This signal is pulled high during reset.	1	1	1	—	1

Table 2-5. External Chip Select

Signal Name	Signal Type	Signal Description	853 Pins	854 Pins	855 Pins	857 Pins	858 Pins
$\overline{CSO}$- $\overline{CS3}$	**Output**	**External Chip Select—**These pins are used as chip selects during external memory cycles.	4	4	4	4	4
(GPIOA0-A3)	Input / Output	**Port A GPIO0-3—**These are General Purpose I/O pins when not configured for EMI use.					

Table 2-6. Host Interface Eight

Signal Name	Signal Type	Signal Description	853 Pins	854 Pins	855 Pins	857 Pins	858 Pins
HD0-HD7	**Input**	**Host Data—**These inputs provide data for the HI registers. These pins are disconnected internally during reset.	8	8	—	8	8
(GPIOB0-B7)	Input / Output	**Port B GPIO 0-7—**These are a General Purpose I/O pins when not configured for host port use.					
HA0-HA2	**Input**	**Host Address—**These inputs provide address selection for HI registers. These pins are disconnected internally during reset.	3	3	—	3	3
(GPIOB8-B10)	Input / Output	**Port B GPIO 10—**These are General Purpose I/O pins when not configured for host port use.					
HRW	**Input**	**Host Read/Write—**When the HI is programmed to interface to a single-data-strobe host bus and the HI function is selected, this signal is the Read/Write input. These pins are disconnected internally during reset.					
($\overline{\text{HRD}}$)	Input	**Host Read Enable—**This signal is the Read Data input when the HI is programmed to interface to a double-data-strobe host bus and the HI function is selected.	1	1	—	1	1
(GPIOB11)	Input / Output	**Port B GPIO 11—**This is a General Purpose I/O pin when not configured for host port use.					
$\overline{\text{HDS}}$	**Input**	**Host Data Strobe—**When the HI is programmed to interface to a single-data-strobe host bus and the HI function is selected, this input enables a data transfer on the HI when $\overline{\text{HCS}}$ is asserted.					
($\overline{\text{HWR}}$)	Input	**Host Write Enable—**This signal is the Write Data input when the HI is programmed to interface to a double-data-strobe host bus and the HI function is selected.	1	1	—	1	1
(GPIOB12)	Input / Output	**Port B GPIO 12—**This is a General Purpose I/O pin when not configured for host port use.					
$\overline{\text{HCS}}$	**Input**	**Host Chip Select—**This input is the chip select input for the Host Interface.This pin is disconnected internally when reset.	1	1	—	1	1
(GPIOB13)	Input / Output	**Port B GPIO 13—**This is a General Purpose I/O pin when not configured for host port use.					

Table 2-6. Host Interface Eight (Continued)

Signal Name	Signal Type	Signal Description	853 Pins	854 Pins	855 Pins	857 Pins	858 Pins
$\overline{\text{HREQ}}$	**Open Drain Output**	**Host Request—**When the HI is programmed for HRMS = 0 functionality, typically used on a single-data-strobe bus, this open drain output is used by HI to request service from the host processor. The HREQ may be connected to an interrupt request pin of a host processor, a transfer request of a DMA controller, or a control input of external circuitry. This pin is disconnected internally when reset.	1	1	—	1	1
($\overline{\text{HTRQ}}$)	Open Drain Output	**Transmit Host Request—**This signal is the output when the HI is programmed for HRMS = 1 functionality and is typically used on a double-data-strobe bus.					
(GPIOB14)	Input / Output	**Port B GPIO 14—**This is a General Purpose I/O pin when not configured for host port use.					
$\overline{\text{HACK}}$	**Input**	**Host Acknowledge—**When the HI is programmed for the HRMS = 0 functionality, typically used on a single-data-strobe bus, this input has two functions: (1) provide a Host Acknowledge signal for DMA transfers, or (2) to control handshaking and provide a Host Interrupt Acknowledge compatible with the MC68000 family processors. This pin is disconnected internally when reset.	1	1	—	1	1
(HRRQ)	Open Drain Output	**Receive Host Request—**This signal is the output when the HI is programmed for HRMS = 1 functionality, and it is typically used on a double-data-strobe bus.					
(GPIOB15)	Input / Output	**Port B GPIO 15—**This is a General Purpose I/O pin when not configured for host port use.					

Table 2-7. Quad Timer Module

Signal Name	Signal Type	Signal Description	853 Pins	854 Pins	855 Pins	857 Pins	858 Pins
TIO0	**Input** / Output	**Timer Input/Output—**This pin can be independently configured to be either a timer input source or an output.	1	1	1	1	1
(GPIOG0)	Input / Output	**Port G GPIO 0—**This is a General Purpose I/O pin when not configured for Timer use.					
TIO1-TIO3	**Input** / Output	**Timer Input/Output—**These pins can be independently configured to be either a timer input source or an output.	3	3	—	3	3
(GPIOG0-G3)	Input / Output	**Port G GPIO 1-3—**These are General Purpose I/O pins when not configured for Timer use.					

Table 2-8. Interrupt and Program Control

Signal Name	Signal Type	Signal Description	853 Pins	854 Pins	855 Pins	857 Pins	858 Pins
$\overline{\text{IRQA}}$	Input	**External Interrupt Request A and B—**These inputs are asynchronous external interrupt requests, indicating an external device is requesting service. A Schmitt trigger input is used for noise immunity. They can be programmed to be level-sensitive or negative-edge-triggered. If level-sensitive triggering is selected, an external pull-up resistor is required or Wired-OR operation.	1	1	1	1	1
$\overline{\text{IRQB}}$	Input		1	1	1	1	1
MODA, MODB, MODC	Input	**Mode Select—**During the bootstrap process, MODA, MODB, and MODC selects one of the eight bootstrap modes. **Mode Select—**During the bootstrap process, the MODE A, B, and C pins select one of the eight bootstrap modes. These pins are sampled at the end of reset.					
(GPIOH0-H2)	Input / Output	**Port H GPIO 0-2—**These are General Purpose I/O pins after the bootstrap process has completed. **Note:** Any time POR and EXTERNAL resets are active the state of MODE A, B and C pins get asynchronous transferred to the SIM Control Register [14:12] ($1FFF08) respectively. These bits determine the mode in which the part will boot up. **Note:** Software and COP resets do not update the SIM Control Register.	3	3	3	3	3

Table 2-8. Interrupt and Program Control (Continued)

Signal Name	Signal Type	Signal Description	853 Pins	854 Pins	855 Pins	857 Pins	858 Pins
$\overline{\text{RESET}}$	Input	**Reset—**This input is a direct hardware reset on the processor. When $\overline{\text{RESET}}$ is asserted low, the device is initialized and placed in the Reset state. A Schmitt trigger input is used for noise immunity. When the $\overline{\text{RESET}}$ pin is deasserted, the initial chip operating mode is latched from the MODA, MODB, and MODC pins. To ensure complete hardware reset, $\overline{\text{RESET}}$ and $\overline{\text{TRST}}$ should be asserted together. The only exception occurs in a debugging environment when a hardware device reset is required and it is necessary not to reset the JTAG/EOnCE module. In this case, assert $\overline{\text{RESET}}$. Do not assert $\overline{\text{TRST}}$.	1	1	1	1	1
$\overline{\text{RSTO}}$	Output	**Reset Output—**This output is asserted on any reset condition (external reset, low voltage, software, or COP).	1	1	1	1	1

Table 2-9. Serial Communication Interface 0

Signal Name	Signal Type	Signal Description	853 Pins	854 Pins	855 Pins	857 Pins	858 Pins
RXDO	**Input**	**Serial Receive Data 0—**This input receives byte-oriented serial data and transfers it to the SCI0 receive shift receiver.	1	1	1	1	1
(GPIOE0)	Input / Output	**Port E GPIO 0—**This is a General Purpose I/O pin capable of being independently programmed as an input or output pin.					
TXDO	**Output(Z)**	**Serial Transmit Data 0—**This signal transmits data from the SCI0 transmit data register.	1	1	1	1	1
(GPIOE1)	Input / Output	**Port E GPIO 1—**This is a General Purpose I/O pin capable of being independently programmed as an input or output pin.					

Table 2-10. Serial Communication Interface 1

Signal Name	Signal Type	Signal Description	853 Pins	854 Pins	855 Pins	857 Pins	858 Pins
RXD1	**Input**	**Serial Receive Data 1—**This input receives byte-oriented serial data and transfers it to the SCI1 receive shift receiver.	1	1	1	1	1
(GPIOE2)	Input / Output	**Port E GPIO 2—**This is a General Purpose I/O pin capable of being independently programmed as an input or output pin.					
TXD1	**Output(Z)**	**Serial Transmit Data 1—**This signal transmits data from the SCI1 transmit data register.	1	1	1	1	1
(GPIOE3)	Input / Output	**Port E GPIO 3—**This is a General Purpose I/O pin capable of being independently programmed as an input or output pin.					

Table 2-11. Enhanced Synchronous Serial Interface 0

Signal Name	Signal Type	Signal Description	853 Pins	854 Pins	855 Pins	857 Pins	858 Pins
STD0	**Output(Z)**	**ESSI Transmit Data 0—**This output pin transmits serial data from the ESSI Transmitter Shift Register.	1	1	1	1	1
(GPIOC0)	Input / Output	**Port C GPIO 0—**This is a General Purpose I/O pin when the ESSI is not in use.					
SRD0	**Input**	**ESSI Receive Data 0—**This input pin receives serial data and transfers the data to the ESSI Transmitter Shift Register.	1	1	1	1	1
(GPIOC1)	Input / Output	**Port C GPIO 1—**This is a General Purpose I/O pin when the ESSI is not is use.					
SCK0	**Input** / Output	**ESSI Serial Clock 0—**This bidirectional pin provides the serial bit rate clock for the transmit section of the ESSI. The clock signal can be continuous or gated and can be used by both the transmitter and receiver in synchronous mode.	1	1	1	1	1
(GPIOC2)	Input / Output	**Port C GPIO 2—**This is a General Purpose I/O pin when the ESSI is not is use.					

Table 2-11. Enhanced Synchronous Serial Interface 0 (Continued)

Signal Name	Signal Type	Signal Description	853 Pins	854 Pins	855 Pins	857 Pins	858 Pins
SC00	**Input** / Output	**ESSI Serial Control Pin 0—**The function of this pin is determined by the selection of either synchronous or asynchronous mode. For asynchronous mode, this pin will be used for the receive clock I/O. For synchronous mode, this pin is used either for transmitter 1 output or for serial I/O flag 0.	1	1	1	1	1
(GPIOC3)	Input / Output	**Port C GPIO 3—**This is a General Purpose I/O pin when the ESSI is not in use.					
SC01	**Input** / Output	**ESSI Serial Control Pin 1—**The function of this pin is determined by the selection of either synchronous or asynchronous mode. for asynchronous mode, this pin is the receiver frame sync I/O. For synchronous mode, this pin is used either for transmitter 2 output or for serial I/O flag 1.	1	1	1	1	1
(GPIOC4)	Input / Output	**Port C GPIO 4—**This is a General Purpose I/O pin when the ESSI is not in use.					
SC02	**Input** / Output	**ESSI Serial Control Pin 2—**This pin is used for frame sync I/O. SC02 is the frame sync for both the transmitter and receiver in synchronous mode and for the transmitter only in asynchronous more. When configured as an output, this pin is the internally generated frame sync signal. When configured as an input, this pin receives an external sync signal for the transmitter, and the receiver in the synchronous operation.	1	1	1	1	1
(GPIOC5)	Input / Output	**Port C GPIO 5—**This is a General Purpose I/O pin capable of being independently programmed as an input or output pin.					

Table 2-12. Enhanced Synchronous Serial Interface 1

Signal Name	Signal Type	Signal Description	853 Pins	854 Pins	855 Pins	857 Pins	858 Pins
STD1	**Output(Z)**	**ESSI Transmit Data (STD1)**—This output pin transmits serial data from the ESSI Transmitter Shift Register.	—	—	—	1	1
(GPIOD0)	Input / Output	**Port D GPIO 0**—This is a General Purpose I/O pin available when the ESSI is not in use.					
SRD1	**Input**	**ESSI Receive Data (SRD1)**—This input pin receives serial data and transfers the data to the ESSI Receive Shift Register.	—	—	—	1	1
(GPIOD1)	Input **/** Output	**Port D GPIO 1**—This is a General Purpose I/O pin available when the ESSI is not in use.					

Table 2-12. Enhanced Synchronous Serial Interface 1 (Continued)

Signal Name	Signal Type	Signal Description	853 Pins	854 Pins	855 Pins	857 Pins	858 Pins
SCK1	**Input** / Output	**ESSI Serial Clock (SCK1)**—This bidirectional pin provides the serial bit rate clock for the transmit section of the ESSI. The clock signal can be continuous or gated and can be used by both the transmitter and receiver in synchronous mode.	—	—	—	1	1
(GPIOD2)	Input / Output	**Port D GPIO 2**—This is a General Purpose I/O pin available when the ESSI is not in use.					
SC10	**Input** / Output	**ESSI Serial Control Pin 0 (SC10)**—The function of this pin is determined by the selection of either synchronous or asynchronous mode. For asynchronous mode, this pin will be used for the receive clock I/O. For synchronous mode, this pin is used either for transmitter1 output or for serial I/O flag 0.	—	—	—	1	1
(GPIOD3)	Input / Output	**Port D GPIO 3**—This is a General Purpose I/O pin available when the ESSI is not in use.					
SC11	**Input** / Output	**ESSI Serial Control Pin 1 (SC11)**—The function of this pin is determined by the selection of either synchronous or asynchronous mode. For asynchronous mode, this pin is the receiver frame sync I/O. For synchronous mode, this pin is used either for transmitter2 output or for serial I/O flag 1.	—	—	—	1	1
(GPIOD4)	Input / Output	**Port D GPIO (4)**—This is a General Purpose I/O pin available when the ESSI is not in use.					
SC12	**Input** / Output	**ESSI Serial Control Pin 2 (SC12)**—This pin is used for frame sync I/O. SC02 is the frame sync for both the transmitter and receiver in synchronous mode and for the transmitter only in asynchronous mode. 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 transmitter (and the receiver in synchronous operation).	—	—	—	1	1
(GPIOD5)	Input / Output	**Port D GPIO 5**—This is a General Purpose I/O pin available when the ESSI is not in use.					

Table 2-13. Serial Peripheral Interface

Signal Name	Signal Type	Signal Description	853 Pins	854 Pins	855 Pins	857 Pins	858 Pins
MISO	**Input** / Output	**SPI Master In / Slave Out—**This serial data pin is an input to a master device and an output from a slave device. The slave device places data on the MISO line a half-cycle before the clock edge the master device uses to latch the data. The driver on this pin can be configured as an open-drain by the SPI's Wired-OR mode (WOM) bit when this pin is configured for SPI operation.	1	1	—	1	1
(GPIOF0)	Input / Output	**Port F GPIO 0—**This is a General Purpose I/O pin capable of being independently programmed as an input or output pin.					
MOSI	Input / **Output(Z)**	**SPI Master Out / Slave In—**This serial data pin is an output from a master device and an input to a slave device. The master device places data on the MOSI line a half-cycle before the clock edge the slave device uses to latch the data. The driver on this pin can be configured as an open-drain by the SPI's Wired-OR mode (WOM) bit when this pin is configured for SPI operation.	1	1	—	1	1
(GPIOF1)	Input / Output	**Port F GPIO 1—**This is a General Purpose I/O pin capable of being independently programmed as an input or output pin.					
SCK	**Input** / Output	**SPI Serial Clock—**This bidirectional pin provides a serial bit rate clock for the SPI. This gated clock signal is an input to a slave device and is generated as an output by a master device. Slave devices ignore the SCK signal unless the $\overline{SS}$ pin is active low. In both 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. The driver on this pin can be configured as an open-drain driver by the SPI's WOM bit when this pin is configured for SPI operation. When using Wired-OR mode, the user must provide an external pull-up device.	1	1	—	1	1
(GPIOF3)	Input / Output	**Port F GPIO 2—**This is a General Purpose I/O pin capable of being independently programmed as an input or output pin.					
$\overline{SS}$	**Input**	**SPI Slave Select—**This input pin selects a slave device before a master device can exchange data with the slave device. $\overline{SS}$ must be low before data transactions and must stay low for the duration of the transaction. The $\overline{SS}$ line of the master must be held high.	1	1	—	1	1
(GPIOF3)	Input / Output	**Port F GPIO 3—**This is a General Purpose I/O pin capable of being independently programmed as an input or output pin.					

Table 2-14. Clock and Phase Lock Loop Signals

Signal Name	Signal Type	Signal Description	853 Pins	854 Pins	855 Pins	857 Pins	858 Pins
EXTAL	Input	**External Crystal Oscillator Input—**This input should be connected to an external crystal. If an external clock source other than a crystal oscillator is used, EXTAL must be tied off.	1	1	1	1	1
CLKO	Output	**Clock Output**—This pin outputs a buffered clock signal. When enabled, this signal is the system clock divided by four.	1	1	1	1	1
XTAL	Input / Output	**Crystal Oscillator Output—**This output connects the internal crystal oscillator output to an external crystal. If an external clock source other than a crystal oscillator is used, XTAL *must* be used as the input and the EXTAL connected to V_{DDA} /2.	1	1	1	1	1

Table 2-15. JTAG/EOnCE Signals

Signal Name	Signal Type	Signal Description	853 Pins	854 Pins	855 Pins	857 Pins	858 Pins
TCK	Input	**Test Clock Input—**This input pin provides a gated clock to synchronize the test logic and to shift serial data to the JTAG/EOnCE port. The pin is connected internally to a pull-down resistor.	1	1	1	1	1
TDI	Input	**Test Data Input—**This input pin provides a serial input data stream to the JTAG/EOnCE port. It is sampled on the rising edge of TCK and has an on-chip pull-up resistor.	1	1	1	1	1
TDO	Output(Z)	**Test Data Output—**This tri-statable output pin provides a serial output data stream from the JTAG/EOnCE port. It is driven in the Shift-IR and Shift-DR controller states, and changes on the falling edge of TCK.	1	1	1	1	1
TMS	Input	**Test Mode Select Input—**This input pin is used to sequence the JTAG TAP controller's state machine. It is sampled on the rising edge of TCK and has an on-chip pull-up resistor.	1	1	1	1	1
$\overline{\text{TRST}}$	Input	**Test Reset—**As an input, a low signal on this pin provides a reset signal to the JTAG TAP controller. To ensure complete hardware reset, $\overline{\text{TRST}}$ should be asserted whenever $\overline{\text{RESET}}$ is asserted. The only exception occurs in a debugging environment when a hardware reset is required and it is necessary not to rest the JTAG/EOnCE module. In this case, assert $\overline{\text{RESET}}$. Do not assert $\overline{\text{TRST}}$.	1	1	1	1	1
$\overline{\text{DE}}$	**Input** / Output	**Debug Enable—**This is an open-drain, bidirectional, active low signal. As an input, it is a means of entering debug mode of operation from an external command controller. As an output, it is a means of acknowledging the chip has entered debug mode.	1	1	1	1	1

Chapter 3
Memory (MEM)

3.1 Introduction

The 56800E provides separate program and data memory areas. The program area has a 21-bit address range, while the data area has a 24-bit address range. Each area has a data width of 16 bits. In the 5685x, the active areas in memory include:

- Up to 40K × 16 bit Program RAM
- Up to 24K × 16 bit Data RAM
- 1K × 16 bit Boot ROM

Off-chip memory expansion capability up to 4MB program and 16MB data with four programmable chip select signals.

3.2 Program Boot ROM

The 5685x has 1K-word × 16-bit on-chip Program ROM. The program ROM contains the bootstrap firmware program able to perform initial loading of the internal program RAM. It is located in Program memory space at locations $1F0000-$1F03FF. The bootstrap program can load any Program RAM segment from an external memory or serial EEPROM. On exiting the reset state the first instruction is fetched from the program ROM ($1F0000) to start execution of the bootstrap program.

The value on the input pins, MODA, MODB, and MODC, when the last active reset source—reset pin, power-on reset, or software reset—deasserts, it will determine which bootstrap mode is entered.

Note: A COP reset via $\overline{\text{RST}}$ does not alter MOD pins of the SIM register since a COP reset by definition occurs unexpectedly during system operation, therefore possibly no longer providing the required MODE inputs.

Note: A software reset via RST_SW does not alter MOD pins of the SIM register, thereby allowing users to reboot in a different mode without altering the hardware.

Some boot modes, specifically those transferring code to internal PRAM for execution, require header data to synchronize the peripheral, define the transfer start address in PRAM, and define the number of words to load. The following four points describe the required format for available boot modes:

1. The four byte ASCII sequence BOOT (mode 1 only)
2. Two words (4 bytes) defining the number of program words to be loaded (modes 0, 1, 4, 5, and 6 only)

3. Two words (4 bytes) starting address where loaded in the program memory (modes 0, 1, 4, 5, and 6 only)
4. The user program (two bytes for each 16-bit program word) on all boot modes

The four-byte string *BOOT* loads *B* first in Boot Mode One. The bytes/words for the remaining data are loaded least significant byte/word first. The boot code is general-purpose and assumes the number of program words and starting address are valid for the user's system. If the values are invalid, unpredictable results will occur. If a reserved mode is specified, the *debughlt* instruction will be executed causing the software to enter an infinite loop.

Once the bootstrap program completes loading the specified number of words, if applicable to that Boot mode, the bootstrap program jumps to the starting address and executes the loaded program. Some of the bootstrap routines reconfigure the memory map by setting the PRAM DISABLE field in the SIM Control Register. Some bootstrap routines also make specific assumptions about the external clock frequency being applied to the part.

3.2.1 Boot Mode 0: Bootstrap From Byte-Wide External Memory

The PRAM DISABLE remains zero, leaving both internal program and data RAM enabled. The bootstrap program loads program memory from a byte-wide memory located at $040000 using CS0 as the chip select, before jumping to the start of the user code.

3.2.2 Boot Mode 1: Bootstrap From SPI

The PRAM DISABLE remains zero, leaving both internal program and data RAM enabled. The bootstrap program loads program memory from a serial EEPROM via the SPI. GPIOF3 is an alternative function of the Slave Select _B ($\overline{SS}$). When configured and programmed, the alternative function can be used as the $\overline{SS}$ output. This mode is compatible with ATMEL AT25xxx and AT45xxx series serial EEPROMs. In order to determine the correct SPI configuration, the first four bytes in the serial memory must be the string *BOOT* in ASCII.

They are: $42, $4F, $4F and $54. If, after trying all three configurations, *BOOT* is not read, the part will move to the DEBUG HALT state. After the string *BOOT*, the data should continue as described in the data sequence above. After loading the user program, GPIOF3 is returned to its power-on reset state—input under peripheral control—and the bootstrap program jumps to the start of the user code. This boot loader assumes the external clock is being applied at a frequency between 2MHz and 4MHz. For some external devices, it enables the PLL during boot loading but always leaves the PLL off when complete.

3.2.3 Boot Mode 2: Normal Expanded Mode

The PRAM DISABLE remains at zero, leaving both internal program and data RAM enabled. No code is loaded. The bootstrap program simply vectors to external program memory location P:$040000 using CS0 as the chip select.

3.2.4 Boot Mode 3: Development Expanded Mode

The PRAM DISABLE remains at one, leaving internal data RAM enabled, but the internal program RAM is disabled. All references to internal program memory space are subsequently directed to external program memory. No code is loaded. The bootstrap program simply vectors to program memory location P:$000000 using CS0 as the chip select.

3.2.5 Boot Mode 4: Bootstrap From Host Port–Single Strobe Clocking

The PRAM DISABLE remains at zero, leaving both internal program and data RAM enabled. The bootstrap program configures the Host Port for single strobe access, loading program memory from the Host Port before jumping to the start of the user code.

3.2.6 Boot Mode 5: Bootstrap From Host Port–Dual Strobe Clocking

The PRAM DISABLE remains at zero, leaving both internal program and data RAM enabled. The bootstrap program configures the Host Port for dual strobe access, loading program memory from the Host Port before jumping tot he start of the user code.

3.2.7 Boot Mode 6: Bootstrap From SCI

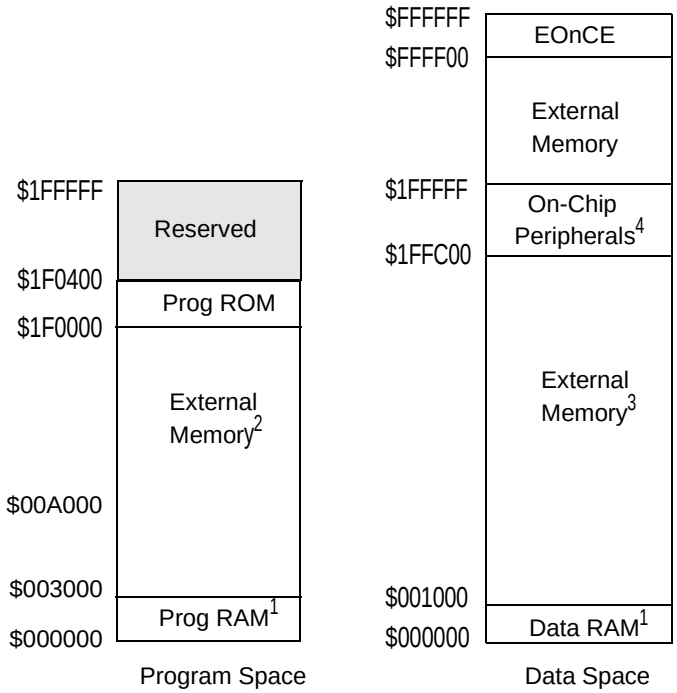
The PRAM DISABLE remains at zero, leaving both internal program and data RAM enabled. It configures the SCI for 38400 baud transfers with a 4MHz or 19200 with 2MHz crystals. It also enables the PLL to operate during the boot process. The bootstrap program then loads program memory from the SCI port and jumps to the start of the user code. External clocking must be between 2MHz and 4MHz. It uses the PLL but leaves it off when complete. The data format is:

- One Start Bit
- Eight-Data Bits
- No Parity Bit
- One Stop Bit
- Flow Control Off

3.2.8 Boot Mode 7: Reserved for Future Use

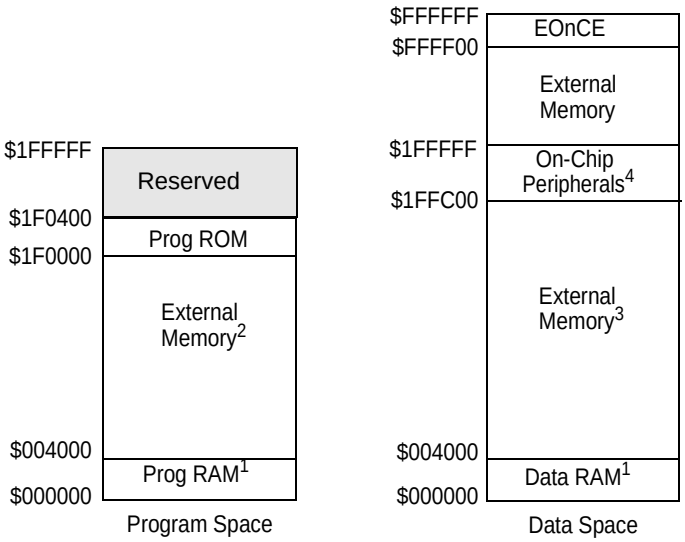
3.3 Memory Maps

Memory maps for each of the five devices are illustrated in **Figure 3-1** through **Figure 3-5.**



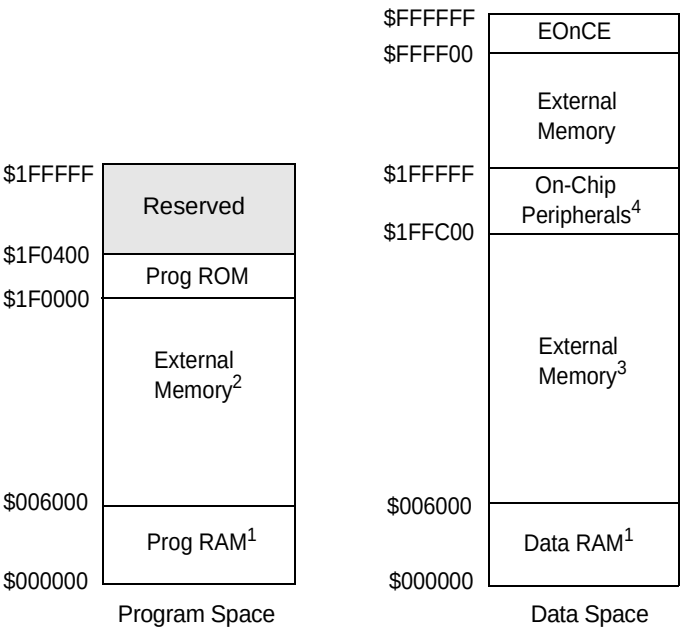
1. Can be disabled resulting in this space being external memory.
2. In operating mode 2, chip select 0 is initially set to $040000 - $07FFFF.
3. In operating mode 0, chip select 0 is initially set to $040000 - $07FFFF.
4. The range of short I/O space is: $1FFFC0 - $1FFFFF.

Figure 3-1. 56853 Memory Map



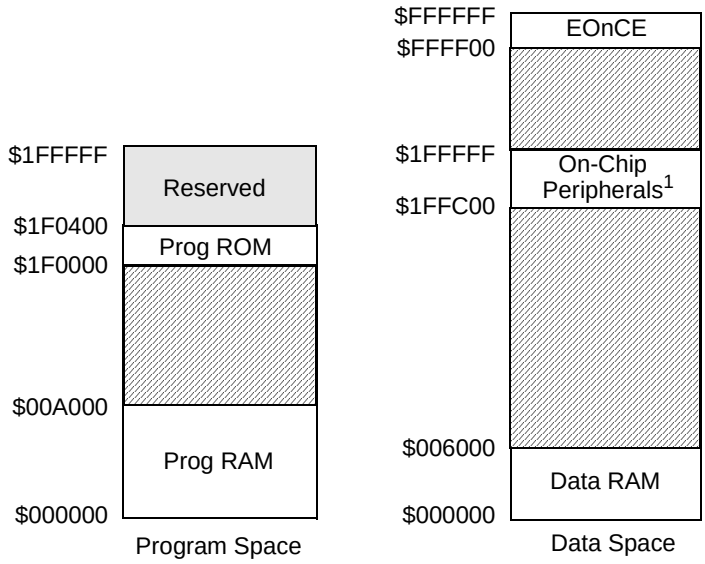
1. Can be disabled resulting in this space being external memory.
2. In operating mode 2, chip select 0 is initially set to $040000 - $07FFFF.
3. In operating mode 0, chip select 0 is initially set to $040000 - $07FFFF.
4. The range of short I/O space is: $1FFFC0 - $1FFFFF.

Figure 3-2. 56854 Memory Map



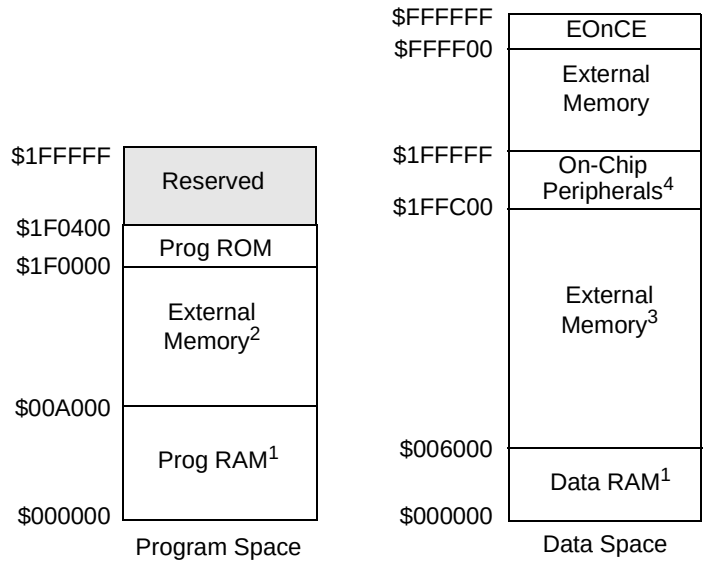
1. Can be disabled, in which case this space is external memory.
2. In operating mode 2, chip select 0 is initially set to $040000 - $07FFFF.
3. In operating mode 0, chip select 0 is initially set to $040000 - $07FFFF.
4. The range of short I/O space is: $1FFFC0 - $1FFFFF.

Figure 3-3. 56855 Memory Map



1. The range of short I/O space is: $1FFFC0 - $1FFFFF.

Figure 3-4. 56857 Memory Map



1. Can be disabled resulting in this space being external memory.
2. In operating mode 2, chip select 0 is initially set to $040000.
3. In operating mode 0, chip select 0 is initially set to $040000 - $07FFFF.
4. The range of short I/O space is: $1FFFC0 - $1FFFFF.

Figure 3-5. 56858 Memory Map

Note: The X (data) address space for all parts is actually 24 bits ($000000 - $FFFFFF) but only 21 bits are brought out externally. The chip selects can be programmed to allow data accesses above $1FFFFF. In this case the data space can be thought of as multiple 2M word pages.

3.3.1 Memory Register Summary

This summary lists all accessible EOnCE Memory mapped registers in the 5685x devices, in **Table 3-1.** The Peripheral System Configurations are listed in **Table 3-15.**

3.3.1.1 Memory Mapped Registers

Table 3-1 lists all of the EOnCE Memory mapped registers in the 5685x devices. Please consult the *DSP56800E Reference Manual* (DSP56800ERM) for complete details of the following table.

Table 3-1. EOnCE Memory Map (EOnCE_BASE = $FFFF00)

Address Offset	Register Acronym	Register Name
Base + $FF	OTX1/ORX1	Transmit Register Upper Word Receive Register Upper Word
Base + $FE	OTX/ORX	Transmit Register Receive Register
Base + $FD	OTXRXSR	Transmit /Receive Status/ Control Reg.
Base + $FC	OCLSR	Core Lock/Unlock Status Register
		Reserved
Base + $A0	OCR	Control Register
Base + $9F	—	Instruction Step Counter
Base + $9E	OSCNTR	Instruction Step Counter
Base + $9D	OSR	Status Register
Base + $9C	OBASE	Peripheral Base Address Register
Base + $9B	OTBCR	Trace Buffer Control Register
Base + $9A	OTBPR	Trace Buffer Pointer Register
Base + $99	—	Trace Buffer Register Stages
Base + $98	OTB	Trace Buffer Register Stages
Base + $97	—	Breakpoint Unit [0] Control Register
Base + $96	OBCR	Breakpoint Unit [0] Control Register
Base + $95	—	Breakpoint 1 Unit [0] Address Register
Base + $94	OBAR1	Breakpoint 1 Unit [0] Address Register
Base + $93	—	Breakpoint 2 Unit [0] Address Register
Base + $92	OBAR2	Breakpoint 2 Unit [0] Address Register
Base + $91	—	Breakpoint 1 Unit [0] Mask Register
Base + $90	OBMSK	Breakpoint 1 Unit [0] Mask Register

Table 3-1. EOnCE Memory Map (EOnCE_BASE = $FFFF00) (Continued)

Address Offset	Register Acronym	Register Name
		Reserved
$8E	OBCNTR	EOnCE Breakpoint Unit [0] Counter
		Reserved
		Reserved
		Reserved
$8A	OESCR	External Signal Control Register
		Reserved

3.3.1.2 Peripheral Mapped Registers

The following tables provide lists of all of the Peripheral Memory mapped registers in the 5685x devices. Each is discussed in this manual.

Table 3-2. System Integration Module Register Address Map (SYS_BASE = $1FFF08) see Chapter 4

Address Offset	Register Acronym	Register Name	Access Type	Chapter Location
Base + $0	SCR	SIM Control Register	Read/Write	Section 4.6.1
Base + $1	SCD1	Software Control Data 1 Register	Read/Write	Section 4.6.2
Base + $2	SCD2	Software Control Data 2 Register	Read/Write	Section 4.6.3

Table 3-3. External Memory Interface Registers Address Map (EMI_BASE = $1FFE40) see Chapter 5

Address Offset	Register Acronym	Register Name	Access Type	Chapter Location
Base + $0	$\overline{CS0}$	Chip Select Address & Block Size	Read/Write	Section 5.6.1
Base + $1	$\overline{CS1}$	Chip Select Address & Block Size	Read/ Write	
Base + $2	$\overline{CS2}$	Chip Select Address & Block Size	Read/Write	
Base + $3	$\overline{CS3}$	Chip Select Address & Block Size	Read/Write	
Base + $8	CSOR_0	Chip Select Options Register 0	Read/Write	Section 5.6.2
Base + $9	CSOR_1	Chip Select Options Register 1	Read/ Write	
Base + $A	CSOR_2	Chip Select Options Register 2	Read/Write	
Base + $B	CSOR_3	Chip Select Options Register 3	Read/Write	
Base + $10	BCR	Bus Control Register	Read/Write	Section 5.6.3

Table 3-4. Clock Generation Module Registers Address Map (CGM_BASE = $1FFF10) see Chapter 6

Address Offset	Register Acronym	Register Name	Access Type	Chapter Location
Base + $0	CGMCR	Control Register	Read/Write	Section 6.6.1
Base + $1	CGMDB	Divide-By Register	Read/Write	Section 6.6.2
Base + $2	CGMTOD	Time-of-Day Register	Read/Write	Section 6.6.3

Table 3-5. Computer Operating Properly Module Registers Address Map (COP_BASE = $1FFFD0) see Chapter 7

Address Offset	Register Acronym	Register Name	Access Type	Chapter Location
Base + $0	COPCTL	Control Register	Read/Write	Section 7.9.1
Base + $1	COPTO	Time-out Register	Read/Write	Section 7.9.2
Base + $2	COPCTR	Counter Register	Read/Write	Section 7.9.3

Table 3-6. Interrupt Control Registers Address Map (ITCN_BASE = $1FFF20) see Chapter 8

Address Offset	Register Acronym	Register Name	Access Type	Chapter Location
Base + $0	IPR0	Interrupt Priority Register 0	Read/Write	Section 8.7.1
Base + $1	IPR1	Interrupt Priority Register 1	Read/Write	Section 8.7.2
Base + $2	IPR2	Interrupt Priority Register 2	Read/Write	Section 8.7.3
Base + $3	IPR3	Interrupt Priority Register 3	Read/Write	Section 8.7.4
Base + $4	IPR4	Interrupt Priority Register 4	Read/Write	Section 8.7.5
Base + $5	IPR5	Interrupt Priority Register 5	Read/Write	Section 8.7.6
Base + $6	IPR6	Interrupt Priority Register 6	Read/Write	Section 8.7.7
Base + $7	IPR7	Interrupt Priority Register 7	Read/Write	Section 8.7.8
Base + $8	IPR8	Interrupt Priority Register 8	Read/Write	Section 8.7.9
Base + $9	VBA	Vector Base Address Register	Read/Write	Section 8.7.10
Base + $A	FIM0	Fast Interrupt Match Register 0	Read/Write	Section 8.7.11
Base + $B	FIVAL0	Fast Interrupt Vector Address Low 0	Read/Write	Section 8.7.12
Base + $C	FIVAH0	Fast Interrupt Vector Address High 0	Read/Write	
Base + $D	FIM1	Fast Interrupt Match Register 1	Read/Write	Section 8.7.13
Base + $E	FIVAL1	Fast Interrupt Vector Low Register 1	Read/Write	Section 8.7.14
Base + $F	FIVAH1	Fast Interrupt Vector High Register 1	Read/Write	
Base + $10	IRQP0	IRQ Pending Register 0	*Read Only*	Section 8.7.15
Base + $11	IRQP1	IRQ Pending Register 1	*Read Only*	
Base + $12	IRQP2	IRQ Pending Register 2	*Read Only*	
Base + $13	IRQP3	IRQ Pending Register 3	*Read Only*	
Base + $14	IRQP4	IRQ Pending Register 4	*Read Only*	
Base + $1A	ICTL	Interrupt Control Register	Read/Write	Section 8.7.16

Table 3-7. Direct Memory Access 0 Register Address Map (DMA0_BASE = $1FFEC0) see Chapter 9

Address Offset	Register Acronym	Register Name	Access Type	Chapter Location
Base + $0	DMA_0_TC	Transfer Control Register	Read/Write	Section 9.6.1
Base + $1	DMA_0_CQS	Circular Queue Size Register	Read/Write	Section 9.6.2
Base + $2	DMA_0_TC	Transfer Count Register	Read/Write	Section 9.6.3
Base + $3	DMA_0_DAL	Destination Address-Low Register	Read/Write	Section 9.6.4
Base + $4	DMA_0_DAH	Destination Address-High Register	Read/Write	Section 9.6.5
Base + $5	DMA_0_SAL	Source Address-Low Register	Read/Write	Section 9.6.6
Base + $6	DMA_0_SAH	Source Address-High Register	Read/Write	Section 9.6.7

Table 3-8. Direct Memory Access 1 Register Address Map (DMA1_BASE = $1FFEC8) see Chapter 9

Address Offset	Register Acronym	Register Name	Access Type	Chapter Location
Base + $0	DMA_1_TC	Transfer Control Register	Read/Write	Section 9.6.1
Base + $1	DMA_1_CQS	Circular Queue Size Register	Read/Write	Section 9.6.2
Base + $2	DMA_1_TC	Transfer Count Register	Read/Write	Section 9.6.3
Base + $3	DMA_1_DAL	Destination Address-Low Register	Read/Write	Section 9.6.4
Base + $4	DMA_1_DAH	Destination Address-High Register	Read/Write	Section 9.6.5
Base + $5	DMA_1_SAL	Source Address-Low Register	Read/Write	Section 9.6.6
Base + $6	DMA_1_SAH	Source Address-High Register	Read/Write	Section 9.6.7

Table 3-9. Direct Memory Access 2 Register Address Map (DMA2_BASE = $1FFED0) see Chapter 9

Address Offset	Register Acronym	Register Name	Access Type	Chapter Location
Base + $0	DMA_2_TC	Transfer Control Register	Read/Write	Section 9.6.1
Base + $1	DMA_2_CQS	Circular Queue Size Register	Read/Write	Section 9.6.2
Base + $2	DMA_2_TC	Transfer Count Register	Read/Write	Section 9.6.3
Base + $3	DMA_2_DAL	Destination Address-Low Register	Read/Write	Section 9.6.4
Base + $4	DMA_2_DAH	Destination Address-High Register	Read/Write	Section 9.6.5
Base + $5	DMA_2_SAL	Source Address-Low Register	Read/Write	Section 9.6.6
Base + $6	DMA_2_SAH	Source Address-High Register	Read/Write	Section 9.6.7

Table 3-10. Direct Memory Access 3 Register Address Map (DMA3_BASE = $1FFED8) see Chapter 9

Address Offset	Register Acronym	Register Name	Access Type	Chapter Location
Base + $0	DMA_3_TC	Transfer Control Register	Read/Write	Section 9.6.1
Base + $1	DMA_3_CQS	Circular Queue Size Register	Read/Write	Section 9.6.2
Base + $2	DMA_3_TC	Transfer Count Register	Read/Write	Section 9.6.3
Base + $3	DMA_3_DAL	Destination Address-Low Register	Read/Write	Section 9.6.4
Base + $4	DMA_3_DAH	Destination Address-High Register	Read/Write	Section 9.6.5
Base + $5	DMA_3_SAL	Source Address-Low Register	Read/Write	Section 9.6.6
Base + $6	DMA_3_SAH	Source Address-High Register	Read/Write	Section 9.6.7

Table 3-11. Direct Memory Access 4 Register Address Map (DMA4_BASE = $1FFEE0) see Chapter 9

Address Offset	Register Acronym	Register Name	Access Type	Chapter Location
Base + $0	DMA_4_TC	Transfer Control Register	Read/Write	Section 9.6.1
Base + $1	DMA_4_CQS	Circular Queue Size Register	Read/Write	Section 9.6.2
Base + $2	DMA_4_TC	Transfer Count Register	Read/Write	Section 9.6.3
Base + $3	DMA_4_DAL	Destination Address-Low Register	Read/Write	Section 9.6.4
Base + $4	DMA_4_DAH	Destination Address-High Register	Read/Write	Section 9.6.5
Base + $5	DMA_4_SAL	Source Address-Low Register	Read/Write	Section 9.6.6
Base + $6	DMA_4_SAH	Source Address-High Register	Read/Write	Section 9.6.7

Table 3-12. Direct Memory Access 5 Register Address Map (DMA5_BASE = $1FFEE8) see Chapter 9

Address Offset	Register Acronym	Register Name	Access Type	Chapter Location
Base + $0	DMA_5_TC	Transfer Control Register	Read/Write	Section 9.6.1
Base + $1	DMA_5_CQS	Circular Queue Size Register	Read/Write	Section 9.6.2
Base + $2	DMA_5_TC	Transfer Count Register	Read/Write	Section 9.6.3
Base + $3	DMA_5_DAL	Destination Address-Low Register	Read/Write	Section 9.6.4
Base + $4	DMA_5_DAH	Destination Address-High Register	Read/Write	Section 9.6.5
Base + $5	DMA_5_SAL	Source Address-Low Register	Read/Write	Section 9.6.6
Base + $6	DMA_5_SAH	Source Address-High Register	Read/Write	Section 9.6.7

Table 3-13. Serial Communication Interface 0 Registers Address Map (SCI0_BASE = $1FFFE0) see Chapter 10

Address Offset	Register Acronym	Register Name	Access Type	Chapter Location
Base + $0	SCI_0_BR	Baud Rate Register	Read/Write	Section 10.9.1
Base + $1	SCI_0_CR	Control Register	Read/Write	Section 10.9.2
Base + $2	SCI_0_CR2	Control Register 2	Read/Write	Section 10.9.3
Base + $3	SCI_0_SR	Status Register	*Read Only*	Section 10.9.4
Base + $4	SCI_0_DR	Data Register	Read/Write	Section 10.9.5

Table 3-14. Serial Communication Interface 1 Registers Address Map (SCI1_BASE = $1FFDF8) see Chapter 10

Address Offset	Register Acronym	Register Name	Access Type	Chapter Location
Base + $0	SCI_1_BR	Baud Rate Register	Read/Write	Section 10.9.1
Base + $1	SCI_1_CR	Control Register	Read/Write	Section 10.9.2
Base + $2	SCI_1_CR2	Control Register 2	Read/Write	Section 10.9.3
Base + $3	SCI_1_SR	Status Register	*Read Only*	Section 10.9.4
Base + $4	SCI_1_DR	Data Register	Read/Write	Section 10.9.5

Table 3-15. Serial Peripheral Interface Registers Address Map (SPI_BASE =$1FFFE8) see Chapter 11

Address Offset	Register Acronym	Register Name	Access Type	Chapter Location
Base + $0	SPSCR	Status and Control Register	Read/Write	Section 11.11.1
Base + $1	SPDSCR	Data Size and Control Register	Read/Write	Section 11.11.2
Base + $2	SPDRR	Data Receive Register	*Read-Only*	Section 11.11.3
Base + $3	SPDTR	Data Transmit Register	*Write-Only*	Section 11.11.4

Note: SPI Registers are not available on the 56855

Table 3-16. Enhanced Synchronous Serial Interface 0 Registers Address Map (ESSI0_BASE = $1FFE20) see Chapter 12

Address Offset	Register Acronym	Register Name	Access Type	Chapter Location
Base + $0	ESSI_0_STX0	Transmit Register 0	*Write-Only*	**Section 12.8.1**
Base + $1	ESSI_0_STX1	Transmit Register 1	*Write-Only*	**Section 12.8.2**
Base + $2	ESSI_0_STX2	Transmit Regsiter 2	*Write-Only*	**Section 12.8.3**
Base + $3	ESSI_0_SRX	Receive Register	*Read-Only*	**Section 12.8.4**
Base + $4	ESSI_0_SSR	Status Register	*Read-Only*	Section 12.8.7
Base + $5	ESSI_0_SCR2	Control Register 2	Read/Write	**Section 12.8.8**
Base + $6	ESSI_0_SCR3	Control Register 3	Read/Write	Section 12.8.9
Base + $7	ESSI_0_SCR4	Control Register 4	Read/Write	**Section 12.8.10**
Base + $8	ESSI_0_STXCR	Transmit Control Register	Read/Write	Section 12.8.11
Base + $9	ESSI_0_SRXCR	Receive Control Register	Read/Write	
Base + $A	ESSI_0_STSR	Time Slot Register	*Write-Only*	**Section 12.8.12**
Base + $B	ESSI_0_SFCSR	FIFO Control / Status Register	*Read-Only*	**Section 12.8.13**
Base + $C	ESSI_0_TSMA	Transmit Slot Mask Register	Read/Write	**Section 12.8.14**
Base + $D	ESSI_0_TSMB	Transmit Slot Mask Register	Read/Write	
Base + $E	ESSI_0_RSMA	Receive Slot Mask Register	Read/Write	**Section 12.8.15**
Base + $F	ESSI_0_RSMB	Receive Slot Mask Register	Read/Write	

Table 3-17. Enhanced Synchronous Serial Interface 1 Registers Address Map (ESSI1_BASE =$1FFE00) see Chapter 12

Address Offset	Register Acronym	Register Name	Access Type	Chapter Location
Base + $0	ESSI_1_STX0	Transmit Register 0	*Write-Only*	**Section 12.8.1**
Base + $1	ESSI_1_STX1	Transmit Register 1	*Write-Only*	**Section 12.8.2**
Base + $2	ESSI_1_STX2	Transmit Regsiter 2	*Write-Only*	**Section 12.8.3**
Base + $3	ESSI_1_SRX	Receive Register	*Read-Only*	**Section 12.8.4**
Base + $4	ESSI_1_SSR	Status Register	*Read-Only*	Section 12.8.7
Base + $5	ESSI_1_SCR2	Control Register 2	Read/Write	**Section 12.8.8**
Base + $6	ESSI_1_SCR3	Control Register 3	Read/Write	Section 12.8.9
Base + $7	ESSI_1_SCR4	Control Register 4	Read/Write	**Section 12.8.10**
Base + $8	ESSI_1_STXCR	Transmit Control Register	Read/Write	Section 12.8.11
Base + $9	ESSI_1_SRXCR	Receive Control Register	Read/Write	
Base + $A	ESSI_1_STSR	Time Slot Register	*Write-Only*	**Section 12.8.12**
Base + $B	ESSI_1_SFCSR	FIFO Control / Status Register	*Read-Only*	**Section 12.8.13**
Base + $C	ESSI_1_TSMA	Transmit Slot Mask Register	Read/Write	**Section 12.8.14**
Base + $D	ESSI_1_TSMB	Transmit Slot Mask Register	Read/Write	
Base + $E	ESSI_1_RSMA	Receive Slot Mask Register	Read/Write	**Section 12.8.15**
Base + $F	ESSI_1_RSMB	Receive Slot Mask Register	Read/Write	

Table 3-18. Quad Timer Registers Address Map (TMR_BASE = $1FFE80) see Chapter 13

Address Offset	Register Acronym	Register Name	Access Type	Chapter Location
Base + $6, $E, $16, $1E	CTRL	Timer Channel Control Register	Read/Write	Section 13.9.1
Base + $7, $F, $17, $F	SCR	Timer Channel Status/Control Reg.	Read/Write	Section 13.9.2
Base + $0, $8, $10, $18	CMP1	Timer Channel Compare Register 1	Read/Write	Section 13.9.3
Base + $1, $9, $11, $19	CMP2	Timer Channel Compare Register 2	Read/Write	Section 13.9.4
Base + $2, $A, $12, $1A	CAP	Timer Channel Capture Register	Read/Write	Section 13.9.5
Base + $3, $B, $13, $1B	LOAD	Timer Channel Load Register	Read/Write	Section 13.9.6
Base + $4, $C, $14, $1C	HOLD	Timer Channel Hold Register	Read/Write	Section 13.9.7
Base + $5, $D, $15, $1D	CNTR	Timer Channel Counter Register	Read/Write	Section 13.9.8

Table 3-19. Time-Of-Day Registers Address Map (TOD_BASE = $1FFFC0) see Chapter 14

Address Offset	Register Acronym	Register Name	Access Type	Chapter Location
Base + $0	TODCS	Control/Status Register	Read/Write	Section 14.6.2
Base + $1	TODCSL	Clock Scaler Load Register	*Read-Only*	Section 14.6.3
Base + $2	TODSEC	Seconds Register	*Read-Only*	Section 14.6.4
Base + $3	TODSAL	Seconds Alarm Register	*Read-Only*	Section 14.6.5
Base + $4	TODMIN	Minutes Register	*Read-Only*	Section 14.6.6
Base + $5	TODMAL	Minutes Alarm Register	*Read-Only*	Section 14.6.7
Base + $6	TODHR	Hour Register	*Read-Only*	Section 14.6.8
Base + $7	TODHAL	Hour Alarm Register	*Read-Only*	Section 14.6.9
Base + $8	TODDAY	Days Register	*Read-Only*	Section 14.6.10
Base + $9	TODDAL	Days Alarm Register	*Read-Only*	Section 14.6.11

Table 3-20. General Purpose Input/Output A Register Map (GPIOA_BASE = $1FFE60) see Chapter 15

Address Offset	Register Acronym	Register Name	Access Type	Chapter Location
Base + $0	GPIO_A_PER	Peripheral Enable Register	Read/Write	Section 15.8.1
Base + $1	GPIO_A_DDR	Data Direction Register	Read/Write	Section 15.8.9
Base + $2	GPIO_A_DR	Data Register	Read/Write	Section 15.8.17
Base + $3	GPIO_A_PUR	Peripheral Enable Register	Read/Write	Section 15.8.25

Table 3-21. General Purpose Input/Output B Register Map (GPIOB_BASE = $1FFE64) see Chapter 15

Address Offset	Register Acronym	Register Name	Access Type	Chapter Location
Base + $0	GPIO_B_PER	Peripheral Enable Register	Read/Write	Section 15.8.2
Base + $1	GPIO_B_DDR	Data DirectionRegister	Read/Write	Section 15.8.10
Base + $2	GPIO_B_DR	Data Register	Read/Write	Section 15.8.18
Base + $3	GPIO_B_PUR	Peripheral Enable Register	Read/Write	Section 15.8.26

Table 3-22. General Purpose Input/Output C Register Map (GPIOC_BASE = $1FFE68) see Chapter 15

Address Offset	Register Acronym	Register Name	Access Type	Chapter Location
Base + $0	GPIO_C_PER	Peripheral Enable Register	Read/Write	Section 15.8.3
Base + $1	GPIO_C_DDR	Data DirectionRegister	Read/Write	Section 15.8.11
Base + $2	GPIO_C_DR	Data Register	Read/Write	Section 15.8.19
Base + $3	GPIO_C_PUR	Peripheral Enable Register	Read/Write	Section 15.8.27

Table 3-23. General Purpose Input/Output D Register Map (GPIOD_BASE = $1FFE6C) see Chapter 15

Address Offset	Register Acronym	Register Name	Access Type	Chapter Location
Base + $0	GPIO_D_PER	Peripheral Enable Register	Read/Write	Section 15.8.4
Base + $1	GPIO_D_DDR	Data DirectionRegister	Read/Write	Section 15.8.12
Base + $2	GPIO_D_DR	Data Register	Read/Write	Section 15.8.20
Base + $3	GPIO_D_PUR	Peripheral Enable Register	Read/Write	Section 15.8.28

Table 3-24. General Purpose Input/Output E Register Map (GPIOE_BASE = $1FFE70) see Chapter 15

Address Offset	Register Acronym	Register Name	Access Type	Chapter Location
Base + $0	GPIO_E_PER	Peripheral Enable Register	Read/Write	Section 15.8.5
Base + $1	GPIO_E_DDR	Data DirectionRegister	Read/Write	Section 15.8.13
Base + $2	GPIO_E_DR	Data Register	Read/Write	Section 15.8.21
Base + $3	GPIO_E_PUR	Peripheral Enable Register	Read/Write	Section 15.8.29

Table 3-25. General Purpose Input/Output F Register Map (GPIOF_BASE = $1FFE74) see Chapter 15

Address Offset	Register Acronym	Register Name	Access Type	Chapter Location
Base + $0	GPIO_F_PER	Peripheral Enable Register	Read/Write	Section 15.8.6
Base + $1	GPIO_F_DDR	Data DirectionRegister	Read/Write	Section 15.8.14
Base + $2	GPIO_F_DR	Data Register	Read/Write	Section 15.8.22
Base + $3	GPIO_F_PUR	Peripheral Enable Register	Read/Write	Section 15.8.30

Table 3-26. General Purpose Input/Output G Register Map (GPIOG_BASE = $1FFE78) see Chapter 15

Address Offset	Register Acronym	Register Name	Access Type	Chapter Location
Base + $0	GPIO_G_PER	Peripheral Enable Register	Read/Write	Section 15.8.7
Base + $1	GPIO_G_DDR	Data DirectionRegister	Read/Write	Section 15.8.15
Base + $2	GPIO_G_DR	Data Register	Read/Write	Section 15.8.23
Base + $3	GPIO_G_PUR	Peripheral Enable Register	Read/Write	Section 15.8.31

Table 3-27. General Purpose Input/Output H Register Map (GPIOH_BASE = $1FFE7C) see Chapter 15

Address Offset	Register Acronym	Register Name	Access Type	Chapter Location
Base + $0	GPIO_H_PER	Peripheral Enable Register	Read/Write	Section 15.8.8
Base + $1	GPIO_H_DDR	Data DirectionRegister	Read/Write	Section 15.8.16
Base + $2	GPIO_H_DR	Data Register	Read/Write	Section 15.8.24
Base + $3	GPIO_H_PUR	Peripheral Enable Register	Read/Write	Section 15.8.32

Table 3-28. Host Interface 8 Registers (HI8_BASE =$1FFFD8) see Chapter 16

Address Offset	Register Acronym	Register Name	Access Type	Chapter Location
Base + $0	HCR	Control Register	Read/Write	Section 16.8.1
Base + $1	HSR	Status Register	Read/Write	Section 16.8.2
Base + $2	HRX	Receive Data Register	*Write-only*	Section 16.8.3
	HTX	Transmit Data Register	*Read-only*	Section 16.8.4

Note: HI8 Registers are not available on the 56855

3.3.2 Interrupt Vectors

The interrupt vectors for the 5685x devices reside in the program memory area. Default addresses of each vector is listed in the Interrupt Controller (ITCN) Chapter, and outlined in **Table 8-3.**

Reset is considered to be the *highest priority interrupt* and takes precedence over all other interrupts. If the reset pin is pulled low, the interrupt controller generates a reset vector address for the core.

Note: The reset vector for the 5685x series devices is $1F0000, the start address of the internal Boot ROM.

Chapter 4
System Integration Module (SIM)

4.1 Introduction

The System Integration Module (SIM) is responsible for several system control functions including:

- Clock generation
- Reset generation
- Power mode control
- Boot mode control
- Memory map control
- External I/O configuration
- Software control

4.2 Features

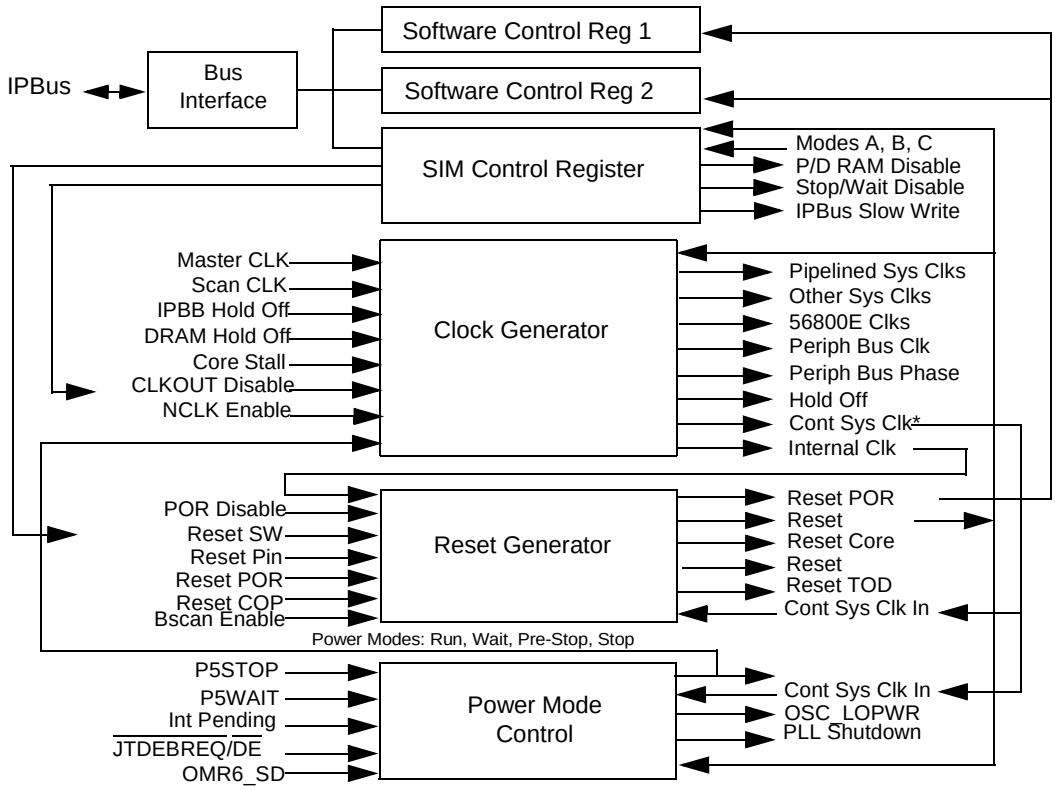
The following list contains distinctions of SIM:

- Five system bus clocks with pipeline hold-off support
  - Data RAM clock with hold-off control
  - IPBus Interface clock with hold-off control
  - System external memory interface four-phase clock with hold-off control
  - 56800E core system clock
  - General purpose clock (both standard and inverted versions)
- Three system clocks for non pipelined interfaces
  - PCLK clock for 56800E core
  - NCLK clock for 56800E core
  - A continuously running system clock
- A peripheral bus (IPBus) clock, both on standard and inverted versions
- An external clock output with disable
- A core stall control used to stall the 56800E core system clock for DMA mastership
- A peripheral bus clock phase indicator
- Three power modes to control power utilization:
  - Stop mode shuts down the 56800E core, system clocks, and the peripheral clock. Stop mode entry can optionally disable PLL and Oscillator providing a choice of lower power vs. a faster restart.

  - Wait mode shuts down the 56800E core, and unnecessary system clock operation.
  - Run mode supports full part operation.
- Controls to enable/disable the 56800E core Wait and Stop instructions
- 32-cycle extended chip internal reset
- 32-cycle extended Time-of-Day reset (responds only to POR reset)
- 32-cycle extended reset for the Clock Generated Module (CGM)
- Software initiated reset
- Controls to redirect internal data and/or program RAM accesses to the external memory interface
- Software Boot mode Control Register, initialized at any reset (except COP reset) from external pins
- A hold off output to coordinate peripheral bus transactions with the system bus pipeline
- Two 16-bit registers reset only by a power-on reset usable for general purpose software control

4.3 Block Diagram

The System Integration Module (SIM) is depicted in **Figure 4-1.**



*Cont SYS_CLK_IN is a synthesized clock tree fed by the CLK_SYS Cont output of SIM

Figure 4-1. System Integration Module

4.4 Signal Description

A description of the System Integration Module (SIM) signals are shown in **Tables 4-1** through **Tables 4-6.**

4.4.1 SIM Interface Signals

Table 4-1. IPBus Signals

Name	Type	Clock Domain	Function
CLK_IPB	Input	—	Peripheral bus clock
RD_DATA_Z	Output	CLK_IPB	Read data (tri-stateable)
WR_DATA	Input	CLK_IPB	Write data
ADDR	Input	CLK_IPB	R/W address (two LSBs of IPBus Address)
RWB	Input	CLK_IPB	Write enable (active low)
$\overline{\text{MODULE\_EN}}$	Input	CLK_IPB	Module enable (active low)

Table 4-2. Clock Generator Inputs/Outputs

Name	Type	Clock Domain	Function
CLK_SYS_DRAM	Output	CLK_MSTR	System clock to data RAM with hold off support
CLK_SYS_IPBB	Output	CLK_MSTR	System clock to IPBus Bridge with hold off support
CLK_SYS_CPUCLK	Output	CLK_MSTR	System clock to 56800E Core with hold off support
CLK_CPU_PCLK	Output	CLK_MSTR	Feeds PCLK input on 56800E Core
CLK_CPU_NCLK	Output	CLK_MSTR	Feeds NCLK input on 56800E Core
CLK_CPU_WCLK	Output	CLK_MSTR	Feeds WRAP_CLK input on 56800E Core
CLK_SYS_GENRI	Output	CLK_MSTR	General purpose system clock with hold off support
CLK_SYS_GENRI_INV	Output	CLK_MSTR	General purpose system clock with hold off support-inverted
CLK_SYS_CONT	Output	CLK_MSTR	Continuous system clock (feeds back into SIM)
CLK_PER_CONT	Output	CLK_MSTR	Peripheral bus clock
CLK_PER_CONT_INV	Output	CLK_MSTR	Inverted peripheral bus clock
CLK_CLKOUT	Output	CLK_MSTR	Output to CLKOUT output pad
PCLK_PHASE	Output	CLK_MSTR	Indicates peripheral clock phase (1=address 0=data)
HOLD OFF	Output	CLK_MSTR	Indicates at least one hold off control is asserted, used to abort peripheral bus transactions
C7WAITST	Output	CLK_MSTR	Indicates to core if it's system clock is to be stalled for reasons other than a core reset (e.g. a hold off or core_stall)
CLK_MSTR	Input	CLK_MSTR	Master input clock from CGM module
CLK_OSC	Input	—	Master clock direct from oscillator bypassing CGM module
CLK_SCAN	Input	—	Scan mode clock
HOLD_DRAM	Input	—	Hold off request from data RAM
HOLD_IPBB	Input	CLK_MSTR	Hold off request from IPBus bridge

Table 4-2. Clock Generator Inputs/Outputs (Continued)

Name	Type	Clock Domain	Function
N1CLKEN	Input	CLK_MSTR	NCLK enable signal from 56800E Core
JHAWKCORETAP_EN	Input	CLK_MSTR	Re synchronized to CLK_SYS_CONT and used to disable CLK_CPU_PCLK after reset if core tap disabled
CORE_STALL	Input	—	Core stall request from SBC for DMA mastership

Table 4-3. Reset Generator Inputs/Outputs

Name	Type	Clock Domain	Function
$\overline{\text{RST\_CORE}}$	Output	CLK_SYS_CONT	Synchronized and extended reset to 56800E Core
$\overline{\text{RST\_PERIPH}}$	Output	CLK_SYS_CONT	Synchronized and extended reset to general peripheral logic
$\overline{\text{RST\_TOD}}$	Output	CLK_SYS_CONT	Synchronized and extended reset to modules using only power on reset (the TOD and COP modules)
$\overline{\text{RST\_CGM}}$	Output	CLK_OSC	Synchronized and extended reset to CGM module
$\overline{\text{RST\_PIN}}$	Input	—	Reset request from external reset pin
$\overline{\text{RST\_POR}}$	Input	—	Reset request from power-on reset module
$\overline{\text{RST\_COP}}$	Input	—	Reset request from COP module

Table 4-4. Register Inputs/Outputs

Name	Type	Clock Domain	Function
MODE_CBA	Input	—	From MODEC, B, A input pads, captured at reset in SIM control register to indicate software boot mode
PRAM_DBL	Output	CLK_IPB	Redirect program RAM accesses to external memory IF
DRAM_DBL	Output	CLK_IPB	Redirect data RAM accesses to external memory IF
STOP_DBL	Output	CLK_IPB	Direct the core to disable the STOP instruction
WAIT_DBL	Output	CLK_IPB	Direct the core to disable the WAIT instruction

Table 4-5. Power Mode Control Inputs/Outputs

Name	Type	Clock Domain	Function
STOPMD	Output	CLK_SYS_CONT	Indicates SIM is in Stop mode
WAITMD	Output	CLK_SYS_CONT	Indicates SIM is in Wait mode
RUNMD	Output	CLK_SYS_CONT	Indicates SIM is in Run mode
OSC_LOPWR	Output	CLK_SYS_CONT	Puts oscillator into low power mode configuration during Stop mode
PLL_SHUTDOWN	Output	CLK_SYS_CONT	Shuts down PLL and puts it into bypass mode when entering Stop mode
P5STOP	Input	CLK_SYS_CONT	Input from Core indicating Stop instruction executed
P5WAIT	Input	CLK_SYS_CONT	Input from Core indicating Wait instruction executed
INT_PEND	Input	CLK_SYS_CONT	Input from INTC indicating interrupt is pending
JTDEBREQ	Input	CLK_SYS_CONT	Input from Core indicating a JTAG debug mode request

Table 4-5. Power Mode Control Inputs/Outputs (Continued)

Name	Type	Clock Domain	Function
$\overline{DE}$	Input	CLK_SYS_CONT	Input from $\overline{DE}$ input pad used to enter OnCE debug mode
OMR6_SD	Input	CLK_SYS_CONT	From core OMR6 register to enable fast stop mode recovery
BSCAN_EBL	Input	—	From external TAP controller indicating boundary scan mode

Table 4-6. Derived Clock Inputs

Name	Type	Clock Domain	Function
CLK_SYS_CONT_IN	Input	CLK_SYS_CONT	Continuous clock fed by synthesized clock tree originating at SIM output CLK_SYS_CONT

4.5 Module Memory Map

The System Integration Module (SIM) contains three programmable 16-bit registers. The address range from $1FFF08 to $1FFF0F is allocated to the SIM. The register is accessed by each of the eight-memory mapped addresses, delineated in **Tables 4-7.** To avoid unpredictable behavior, reserved registers *must not* be written. A read from an address without an associated register returns unknown data.

Table 4-7. System Integration Module Memory Map (SIM_BASE = $1FFF08)

Address Offset	Register Acronym	Register Name	Access Type	Chapter Location
Base + $0	SCR	SIM Control Register	Read/Write	Section 4.6.1
Base + $1	SCD1	Software Control Register 1	Read/Write	Section 4.6.2
Base + $2	SCD2	Software Control Register 2	Read/Write	Section 4.6.3

Add. Offset	Register Name		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
$0	SCR	R	0	BOOT MODE			CHIP REV				0	EOnCE EBL	CLKOUT DBL	PRAM DBL	DRAM DBL	SW DBL	STOP DBL	WAIT DBL
		W																
$1	SCID1	R	SOFTWARE CONTROL DATA 1															
		W																
$2	SCD2	R	SOFTWARE CONTROL DATA 2															
		W																

R 0 Read as 0
W Reserved

Figure 4-2. SIM Register Map Summary

4.6 Register Descriptions (SYS_BASE = $1FFF08)

4.6.1 SIM Control Register (SCR)

Base + $0	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	0	BOOT MODE			CHIP REV				0	EOnCE EBL	CLKOUT DBL	PRAM DBL	DRAM DBL	SW RST	STOP DBL	WAIT DBL
Write																
Reset	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0

Figure 4-3. SIM Control Register (SCR)

See Programmer's Sheet on Appendix page B-8

4.6.1.1 Reserved—Bit 15

This bit is reserved or not implemented. It is read as 0, but it cannot be modified by writing.

4.6.1.2 Boot Mode—Bits 14–12

These boot mode bits replace the mode bits in the 56800E OMR.

This field is set to the value on the input pins, MODA, MODB, and MODC, when the last active reset source—reset pin, power-on reset—deasserts. Synchronous reset (software or COP) do not update Boot mode assuming applications software will set the desired field value prior to the reset. The core always begins execution after reset from the base on the on-chip ROM. The software in ROM will perform one of the several boot actions based on the value of Boot mode.

Note: A COP reset via $\overline{\text{COP\_RST}}$ does not alter these registers since a COP reset by definition occurs unexpectedly during system operation, therefore possibly no longer providing the required MODE inputs.

Note: A software reset via $\overline{\text{RST\_SW}}$ does not alter these registers, thereby allowing users to reboot in a different mode without altering the hardware.

The 56800E core begins execution from the base of the on-chip ROM. The software in ROM will perform one of several boot actions based on the value of BOOT MODE.

Some Boot modes, specifically those transferring code to internal PRAM for execution, require header data to synchronize the peripheral, define the transfer start address in PRAM, and define the number of words to load. The following four points describe the required format for boot mode:

1. The four byte ASCII sequence BOOT (mode 1 only)
2. Two words (4 bytes) defining the number of program words to be loaded (modes 0, 1, 4, 5, and 6 only)

3. Two word (4 bytes) starting address where loaded in the program memory (modes 0, 1, 4, 5, and 6 only)
4. The user program (two bytes for each 16-bit program word) on all boot modes

In Boot mode 1, the four-byte string BOOT loads B first in Boot Mode One. The bytes/words for the remaining data are loaded least significant byte/word first. The boot code is general-purpose, and assumes the number of program words and starting address are valid for the users system. If the values are invalid, unpredictable results will occur. If a reserved mode is specified, the *debughlt* instruction will be executed causing the software to enter an infinite loop.

Once the bootstrap program completes loading the specified number of words, if applicable to that boot mode, the bootstrap program jumps to the starting address and executes the loaded program. Some of the bootstrap routines reconfigure the memory map by setting the PRAM DISABLE field. Some bootstrap routines also make specific assumptions about the external clock frequency being applied to the part.

4.6.1.2.1 Boot Mode 0: Bootstrap From Byte-Wide External Memory

The PRAM DISABLE remains zero, leaving both internal program and data RAM enabled. The bootstrap program loads program memory from a byte-wide memory located at $040000 using CS0 as the chip select, before jumping to the start of the user code.

4.6.1.2.2 Boot Mode 1: Bootstrap From SPI

The PRAM DISABLE remains zero, leaving both internal program and data RAM enabled. The bootstrap program loads program memory from a serial EEPROM via the SPI. GPIOF3 is an alternative function of the $\overline{SS}$, which when configured and programmed, can be used as the $\overline{SS}$ output. This mode is compatible with ATMEL AT25xxx and AT45xxx series serial EEPROMs. In order to determine the correct SPI configuration, the first four bytes in the serial memory must be the string *BOOT* in ASCII. They are: $42, $4F, $4F and $54. If, after trying all three configurations, *BOOT* is not read, the part will move to the DEBUG HALT state. After the string *BOOT*, the data should continue as described in the data sequence above. After loading the user program, GPIOF3 is returned to its power-on reset state—input under peripheral control—and the bootstrap program jumps to the start of the user code. This boot loader assumes the external clock is being applied at a frequency between 2MHz and 4MHz. For some external devices, it enables the PLL during boot loading but always leaves the PLL off when complete.

4.6.1.2.3 Boot Mode 2: Normal Expanded Mode

The PRAM DISABLE remains at zero, leaving both internal program and data RAM enabled. No code is loaded. The bootstrap program simply vectors to external program memory location P:$040000 using CS0 as the chip select.

4.6.1.2.4 Boot Mode 3: Development Expanded Mode

The PRAM DISABLE is set to one, leaving internal data RAM enabled, but the internal program RAM is disabled. All references to internal program memory space are subsequently directed to external program memory. No code is loaded. The bootstrap program simply vectors to program memory location P:$000000 using CS0 as the chip select.

4.6.1.2.5 Boot Mode 4: Bootstrap From Host Port–Single Strobe Clocking

The PRAM DISABLE remains at zero, leaving both internal program and data RAM enabled. The bootstrap program configures the Host Port for single strobe access, loading program memory from the Host Port before jumping to the start of the user code.

Note: This mode becomes a reserved mode on parts lacking a host port interface.

4.6.1.2.6 Boot Mode 5: Bootstrap From Host Port–Dual Strobe Clocking

The PRAM DISABLE remains at zero, leaving both internal program and data RAM enabled. The bootstrap program configures the Host Port for dual strobe access, loading program memory from the Host Port before jumping to the start of the user code.

Note: This mode becomes a reserved mode on parts lacking a host port interface.

4.6.1.2.7 Boot Mode 6: Bootstrap From SCI

The PRAM DISABLE remains at zero, leaving both internal program and data RAM enabled. It configures the SCI for 38400 baud transfers with a 4MHz or 19200 with a 2MHz crystal. It also enables the PLL to operate during the boot process. The bootstrap program then loads program memory from the SCI port and jumps to the start of the user code. External clocking must be at 2MHz or 4MHz. It uses the PLL but leaves it off when complete. The data format is:

- One start bit
- Eight-data bits
- No parity bit
- One Stop bit
- Flow Control off

4.6.1.2.8 Boot Mode 7: Reserved for Future Use

4.6.1.3 Chip Revision (CHIP REV)—Bits 11–8

Static read only value (currently 0001) indicating chip revision. This is intended to be used by applications software to facilitate support of backwards compatibility. Changes to the programming interface can be tied to their associated chip revision.

4.6.1.4 Reserved—Bit 7

This bit is reserved or not implemented. It is a read/write bit, initializing to one.

4.6.1.5 Enhanced OnCE Enable (EOnCE_EBL)—Bit 6

The clock for the 56800E core On-Chip Emulator debug features is normally enabled with the core TAP interface. Setting this bit continuously enables the clock. This powers the core to perform I/O with its mapped EOnCE registers without regard for the TAP state. This is useful when doing real time debugging.

- 0 = EOnCE clock to core is enabled only when core TAP is enabled
- 1 = EOnCE clock to core is always enabled

4.6.1.6 CLKOUT Disable (CLKOUT_DBL)—Bit 5

- 0 = CLKOUT (CLKO) output pin presents CLK_MSTR/8 (this is half the peripheral bus clock frequency)
- 1 = CLKOUT (CLKO) output pin presents static zero

4.6.1.7 Program RAM Disable (PRAM_DBL)—Bit 4

- 0 = Internal program RAM enabled
- 1 = Internal program RAM disabled and accesses redirected to external memory

4.6.1.8 Data RAM Disable (DRAM_DBL)—Bit 3

Note: This field replaces the EX bit in the core OMR.

- 0 = Internal data RAM enabled
- 1 = Internal data RAM disabled and accesses redirected to external memory

4.6.1.9 Software Reset (SW_RESET)—Bit 2

Writing 1 to this field results in the part to reset.

4.6.1.10 Stop Disable (STOP_DBL)—Bit 1

- 0 = The Stop mode is entered when the core executes a Stop instruction
- 1 = The core Stop instruction will not cause entry into the Stop mode

4.6.1.11 Wait Disable (WAIT_DBL)—Bit 0

- 0 = The Wait mode is entered when the core executes a Wait instruction
- 1 = The core Wait instruction will not cause entry into the Wait mode

4.6.2 SIM Software Control Data 1 (SCD1)

Base + $1	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	SOFTWARE CONTROL DATA 1															
POR	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Reset	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

Figure 4-4. SIM Software Control Data 1 Register (SCD1)

See Programmer's Sheet on Appendix page B-9

4.6.2.1 Software Control Data 1 (SCD1)—Bits 15–0

This register is reset only by the Power-On Reset (POR). It has no part specific functionality. It is intended for use by software developers to contain data to be unaffected by the other reset sources:

- Reset pin
- Software reset
- COP reset

4.6.3 SIM Software Control Data 2 (SCD2)

Base + $2	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	SOFTWARE CONTROL DATA 2															
POR	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Reset	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

Figure 4-5. SIM Software Control Data 2 Register (SCD2)

See Programmer's Sheet on Appendix page B-9

4.6.3.1 Software Control Data 2 (SCD2)—Bits 15–0

This register is reset only by the Power-on Reset (POR). It has no part specific functionality. It is intended for software developers to contain data to be unaffected by the other reset sources:

- Reset pin
- Software reset
- COP reset

4.7 Clock Generation Concepts

The 568500E system bus is pipelined. The data cycle occurs two system clock cycles after the address cycle with a *between* cycle between the two. The SIM also supports two system bus masters, the core and multi-channel DMA. The SIM contains features to maintain the integrity of this continuous pipeline during the operation of the system. A hold-off mechanism is furnished to provide system bus slaves extra clock cycles when needed to maintain synchronization with the pipeline. A core stall mechanism is provided to keep the 56800E core's internal pipeline from advancing when the DMA is master.

The peripheral bus clock has no hold-off, or stall. It runs at one-half the frequency of the system bus. The peripheral bus reads and writes are single cycle. IPBB is the only master on the peripheral bus.

A Wait state mechanism permits peripheral bus transactions to take extra cycles when required by slower peripherals.

The IPBus Bridge (IPBB) coordinates the system and peripheral buses. The IPBB presents hold-off requests to the system bus pipeline when the system bus transactions generate multiple peripheral bus transactions, or the peripheral is programmed to generate Wait states.

All generated clocks are true for the first-half period and false for the second-half period. Any mechanism inhibiting a clock such as hold-off, or core stall, causes the clock to remain low during the entire period. The CLK_MSTR input can be fed by the PLL output, by the oscillator running with a crystal, or by the oscillator input being clocked externally. The system bus clock frequency is always CLK_MSTR/2. The peripheral bus clock frequency is always CLK_MSTR/4. The CLKOUT frequency is always CLK_MSTR/8.

4.8 Generated Clocks

A description of the System Integrated Module (SIM) clock signals is shown in **Table 4-8.**

Table 4-8. SIM Clock Signals

Clock Output	Frequency	Enable Condition	Used By
CLK_SYS_DRAM	CLK_MSTR/2	((~HOLD OFF)¾ HOLD_DRAM) and (~Stop mode)	Data RAM
CLK_SYS_IPBB	CLK_MSTR/2	((~HOLD OFF) ¾HOLD_IPBB) and (~Stop mode)	IPBus Bridge
CLK_SYS_CPUCLK	CLK_MSTR/2	(~c7WAITST) and Run mode	Core
CLK_CPU_PCLK	CLK_MSTR/2	(~Stop mode) and ((~RST_$\overline{\text{CORE}}$)¾ JHAWKCORETAP_EN ¾ EOnCE EBL **Note:** JHAWKCORETAP_EN is re synchronized before use	Core EOnCE clock
CLK_CPU_NCLK	CLK_MSTR/2	(~Stop mode) and (~RST_$\overline{\text{CORE}}$)¾ N1CLKEN)	Core EOnCE clock
CLK_CPU_WCLK	CLK_SCAN	Always Enabled	Core scan clock
CLK_SYS_GENRL	CLK_MSTR/2	(~HOLD OFF) and (~Stop mode)	SIM, Program RAM, SBC, SAD, ROM
CLK_SYS_GENRL_INV	CLK_MSTR/2	(~HOLD OFF), (~Stop mode), and (TMODE_BIST)	RAM and ROM BIST
CLK_CPU_CONT	CLK_MSTR/2	Always Enabled	SIM
CLK_PER_CONT	CLK_MSTR/4	(~Stop mode)	Peripherals
CLK_PER_CONT_INV	CLK_MSTR/4	(~Stop mode)	EMI
CLK_CLKOUT	CLK_MSTR/8	(~CLKOUT Disable)	Output pad CLKOUT
PCLK_PHASE	—	Identical to CLK_PER_CONT, but phase shifted to rise earlier	DMA, IPBB
HOLD OFF	—	HOLD_DRAM ¾ HOLD_IPBB	PAD, IPBB
C7WAITST	—	(HOLD OFF¾ CORE-STALL) and RST_$\overline{\text{CORE}}$	Core

4.9 Power Mode Controls

The Power Mode Control module controls movement between the three power modes supported by the core:

1. Run mode provides full functionality
2. Functions disabled in Wait mode:
   — execution of the core
   — any unnecessary system clocks
3. Functions disabled in Stop mode:
   — 56800E core
   — all system clocks
   — peripheral bus clock
   — PLL optionally
   — OSC optionally

All system clocks continue running in the Wait mode, allowing the DMA to operate in the Wait mode. The time-based clock generated by the oscillator and Clock Generation Module (CGM) are not affected by Low Power modes. Time based functions such as a Time-of-Day (TOD) module and Computer Operating Properly (COP) module must be individually disabled for maximum low-power effects. Likewise, Power Mode Controls do not affect pull-up/pull-down resistor enabling. Power loss through input and bidirectional I/O cell pull-up/pull down resistors can be eliminated by disabling the resistor in the software where supported, or in the case of bidirectional I/O, by putting the cell in an output state and avoiding external contention.

When the core executes a Stop or Wait instruction it will wait until any stall or hold-off activity is completed (c7WAITST has deasserted) then it asserts the p5STOP or p5WAIT SIM input and the SIM will enter the corresponding Low Power mode. The SIM Control register also contains Stop and Wait disable bits and feed the core. When asserted, these cause the core to ignore Stop and Wait instructions, not assert p5STOP or p5WAIT.

Recovery from Stop or Wait mode to Run mode occurs if there is a pending enabled interrupt (INT_PEND input asserts), or if there is a Debug mode request from the core due to a JTAG initiated Debug mode entry request (JTDEBREQ input asserts), or if there is a Debug mode entry request from the $\overline{\text{DE}}$ input pin ($\overline{\text{DE}}$ asserts). These three inputs are asynchronous so they are metastabilized and re-timed to the system clock domain before use.

The SIM has special control relationships with both the Oscillator (OSC) module and the Phase Locked Loop (PLL) module. By default, the SIM provides an extreme low power Stop mode (when OMR6_SD set to zero), by shutting down the PLL and, if possible, the oscillator output amplifiers. Alternatively (when OMR6_SD set to one), the SIM supports a fast Stop mode recovery and does not affect the state of the PLL or oscillator when the Stop mode is entered.

Extreme low power Stop mode works in this manner: upon entering the Stop mode, the SIM asserts its PLL_SHUTDOWN output causing the PLL to be disabled and bypassed. One cycle later, it asserts its OSC_LOPWR output. This feeds the LOW_PWR_MODE input of the OSC. When the TOD clock prescaler in the OSC module is used (TOD_SEL bit in CGM Control register is zero), and OSC_LOPWR is asserted high, the OSC module shuts off its output clock amplifiers for maximum power savings. When the CGM TOD clock prescaler is used (TOD_SEL bit is one), OSC_LOPWR is ignored because the CGM depends on clocking from OSC to generate the TOD clock.

When a fast Stop mode recovery is applied (the OMR6_SD bit in the core is set) neither OSC_LOPWR nor PLL_SHUTDOWN will assert during the Stop mode entry. In this case, the Stop mode entry leaves the clock generation system alone. When there is a return to the Run mode, the clock (PLL based or direct), will be just as it was when Stop was entered, avoiding any need to wait for PLL lock.

The SIM does not automatically restart and engage the PLL upon recovery from extreme Low Power mode. This responsibility is left to the applications software. Refer to the documentation of the Oscillator module for details on its Low Power mode input.

A final note on timing: Entry into either the Wait or Stop modes occurs at the next system clock edge after p5STOP or p5WAIT asserts. Their disabling effect on clock generation will start one clock cycle after that. The timing of entry into Low Power modes, especially the Stop mode, is critical and is managed by a state machine in the Power Mode Control module. This module ensures while in the Stop mode entry, PLL_SHUTDOWN asserts and the PLL is disabled and in Bypass mode before OSC_LOPWR asserts possibly shutting off the clock input to the PLL.

4.10 Resets

The SIM supports four sources of reset. The two asynchronous sources are the external reset pin and the Power-On Reset (POR). The two synchronous sources are the software reset, generated within the SIM itself, and the COP reset. The reset generation module has two reset detectors. A chip internal reset is detected when any of the sources assert. A POR reset is detected only when the POR input asserts. The detectors remain asserted until the last active reset source deasserts. The chip-internal reset detector output is the primary reset used within the SIM. The only exception is the Software Control registers, reset by the POR reset detector and the Boot mode field in the SIMCTL register, reset by the mode reset detector.

The SIM generates four reset outputs. All are active low. These all are activated by one of the two detectors but remain asserted for 32-system clock cycles after the detector deasserts. This permits the SIM to generate 32-system clock cycles of continuous clocking to the part while the reset remains asserted. This is required to clear synchronous resets within the 56800E core and elsewhere in the part. The $\overline{\text{RST\_CORE}}$, $\overline{\text{RST\_PERIPH}}$, and $\overline{\text{RST\_CGM}}$ outputs are activated by the chip-internal reset detector and the $\overline{\text{RST\_TOD}}$ output is activated by the POR reset detector. The $\overline{\text{RST\_CORE}}$ output is used to reset the core. The $\overline{\text{RST\_TOD}}$ output is used to reset all controls used to configure time-of-day clock to retain their value once the part is powered on. See TOD, COP, OSC chapters. The $\overline{\text{RST\_CGM}}$ is re-timed to the OSC_CLK and is used to reset the CGM module, in turn using the oscillator clock directly. The $\overline{\text{RST\_PERIPH}}$ reset is used to reset everything else.

Standards require the part to be held in reset during boundary scan operations. When the BSCAN_EBL input is asserted, all resets used within the SIM and all reset outputs of the SIM will go to their active asserted state. This prevents accidental damage due to random inputs applied during boundary scan testing.

Power modes discussed in **Section 4.9** affect reset function. COP reset is not honored if the OSC module is in its Low Power mode since the OSC module shuts off all clocking to the part and without a clock, the SIM can't see the synchronous COP reset. COP reset will be honored immediately upon return to Run mode when clocks resume. There are two ways to permit a COP reset to be honored in Stop mode:

1. Either set OMR6_SD (core register OMR6 bit SD), or
2. Set TOD_SEL in the CGM control register

Setting OMR6_SD configures for fast Stop mode recovery. Setting TOD_SEL selects the CGM's time-of-day clock prescaler for use rather than the one in the OSC module. Either, or both, prevent the OSC from shutting off its clock outputs. In turn, this provides an active CLK_MSTR input to the SIM, allowing an immediate COP reset. The Software Reset is only operable in Run mode when the CPU can write to the SIM Control register to activate Software Reset. The Low Power modes and controls are explained in detail in the next chapter.

Chapter 5
External Memory Interface (EMI)

5.1 Introduction

The External Memory Interface (EMI) provides an interface allowing 56800E core to utilize external asynchronous memory. The EMI for the 56800E core operates from the system bus.

The 56800E core EMI is implemented as a core bus peripheral. Data can be transferred through the EMI to the core directly.

The EMI described in this document is intended to be interfaced to 16-bit wide external memory. External arrays may be implemented either using single 16-bit wide parts or pairs of 8-bit wide memories. An external data space memory interface to the 56800E core accommodating single 8-bit wide external data memories could be implemented (at a substantial performance penalty) with appropriate programming of the CSOR register(s).

5.2 Features

The External Memory Interface supports the following general characteristics:

- Can convert any internal bus memory request to a request for external memory
- Can manage multiple internal bus requests for external memory access
- Has up to four $\overline{\text{CS}}n$ configurable outputs for external device decoding
  - — each $\overline{\text{CS}}$ can be configured for Program or Data space
  - — each $\overline{\text{CS}}$ can be configured for Read only, Write only, or Read/Write access
  - — each $\overline{\text{CS}}$ can be configured for the number of wait states required for device access
  - — each $\overline{\text{CS}}$ can be configured for the size and location of its activation
  - — each $\overline{\text{CS}}$ is independently configured for setup and hold timing controls for both read and write

5.3 Functional Description

The 56800E core architecture contains three separate buses for access to memory/peripherals. The EMI attaches to all of these buses and provides an interface to external memory over a single external bus. The EMI serializes these internal requests to external memory in a manner avoiding conflicts and contention.

5.3.1 Core Interface Detail

Managing the core access to the external memory consists of four issues: (Please refer to **Figure 5-1.**)

1. Any of the three buses can request external access at any time. This means the EMI can potentially have three requests it must be completed before the core can proceed. The EMI must hold-off further execution of the core until it can serialize the requests over the external bus. This provides *simultaneous* data for all buses to the core for read operations.
2. There may be a mixture of read and write requests on the core buses. For instance, the program memory bus may request a read operation while the primary data bus (XAB1) is requesting a write operation.
3. The primary data bus (XAB1) may request an 8-bit transfer. This request must access the appropriate external byte.
4. The primary data bus (XAB1) may request a 32-bit transfer. This request requires two accesses of the external bus. The EMI must hold-off further core execution until all 32 bits have been transferred. This action may happen in conjunction with item one above.

5.4 Block Diagram

A simplified block diagram illustrating the connections to the EMI is illustrated in **Figure 5-1.** The left side of the figure shows connections to the 56800E core buses and clocks. All available external EMI signals are shown on the right side of the figure. In some cases, pin count restrictions may limit the number of EMI signals brought out of the package.

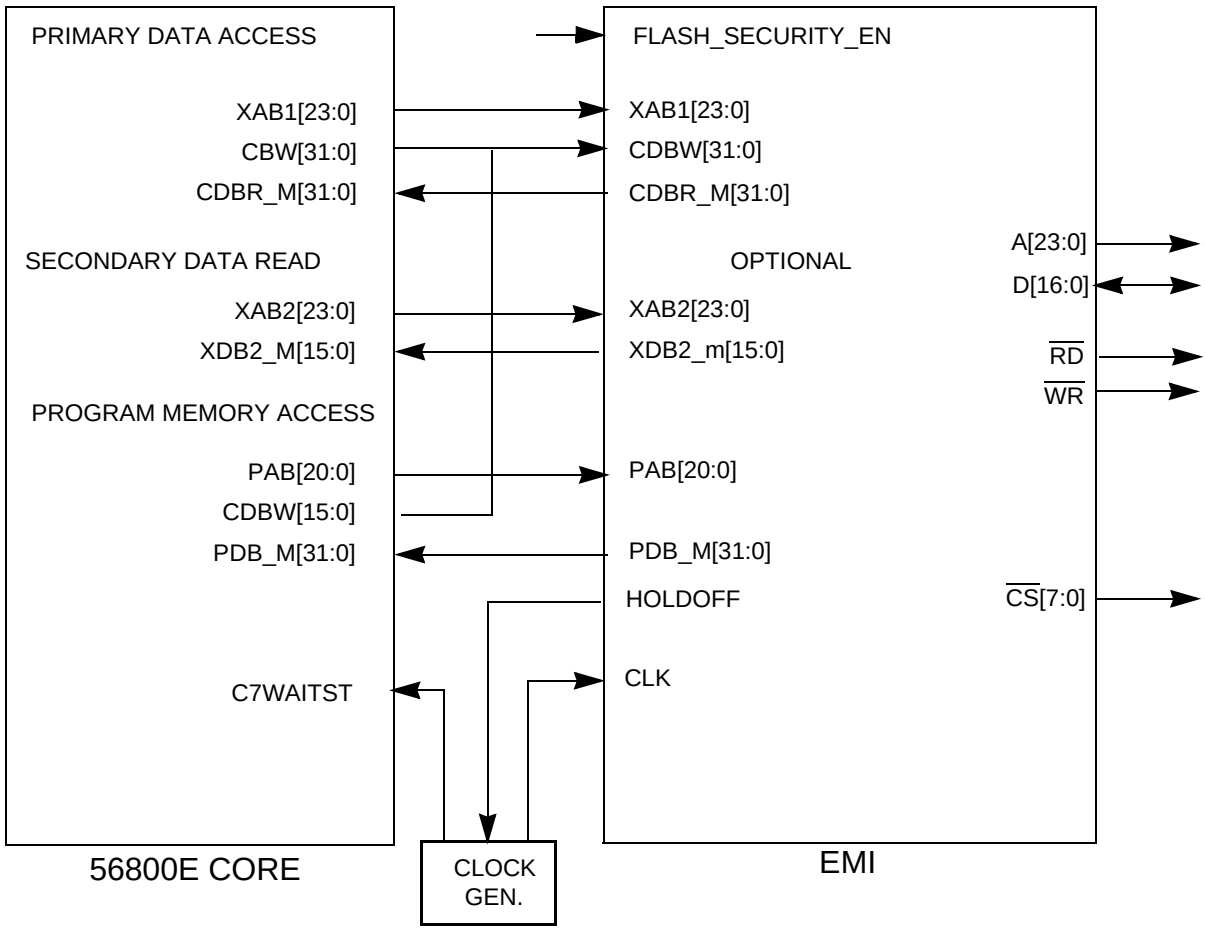


Figure 5-1. EMI Block Diagram

5.5 Module Memory Map

The address of a register is the sum of a base address and an address offset. The base address is defined at the device level. Registers are summarized in **Table 5-1.**

Table 5-1. EMI Module Memory Map (EMI_BASE = $1FFE40)

Address Offset	Register Acronym	Register Name	Chapter Location
Base + $0	CSBAR0	Chip Select Base Address Register 0	Section 5.6.1
Base + $1	CSBAR1	Chip Select Base Address Register 1	
Base + $2	CSBAR2	Chip Select Base Address Register 2	
Base + $3	CSBAR3	Chip Select Base Address Register 3	
Base + $8	CSOR0	Chip Select Option Register 0	Section 5.6.2
Base + $9	CSOR1	Chip Select Option Register 1	
Base + $A	CSOR2	Chip Select Option Register 2	
Base + $B	CSOR3	Chip Select Option Register 3	
Base + $10	CSTC0	Chip Select Timing Control Register 0	Section 5.6.3
Base + $11	CSTC1	Chip Select Timing Control Register 1	
Base + $12	CSTC2	Chip Select Timing Control Register 2	
Base + $13	CSTC3	Chip Select Timing Control Register 3	
Base + $18	BCR	Bus Control Register	Section 5.6.4

Add. Offset	Register Name		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
$0	CSBAR0	R	ADR 23	ADR 22	ADR 21	ADR 20	ADR 19	ADR 18	ADR 17	ADR 16	ADR 15	ADR 14	ADR 13	ADR 12	BLKSZ			
		W																
$1	CSBAR1	R	ADR 23	ADR 22	ADR 21	ADR 20	ADR 19	ADR 18	ADR 17	ADR 16	ADR 15	ADR 14	ADR 13	ADR 12	BLKSZ			
		W																
$2	CSBAR2	R	ADR 23	ADR 22	ADR 21	ADR 20	ADR 19	ADR 18	ADR 17	ADR 16	ADR 15	ADR 14	ADR 13	ADR 12	BLKSZ			
		W																
$3	CSBAR3	R	ADR 23	ADR 22	ADR 21	ADR 20	ADR 19	ADR 18	ADR 17	ADR 16	ADR 15	ADR 14	ADR 13	ADR 12	BLKSZ			
		W																
$8	CSOR0	R	RWS					BYTE_EN		R/W		PS/DS		WWS				
		W																
$9	CSOR1	R	RWS					BYTE_EN		R/W		PS/DS		WWS				
		W																
$A	CSOR2	R	RWS					BYTE_EN		R/W		PS/DS		WWS				
		W																
$B	CSOR3	R	RWS					BYTE_EN		R/W		PS/DS		WWS				
		W																
$10	CSTC0	R	WWSS		WWSH		RWSS		RWSH		0	0	0	0	0	MDAR		
		W																
$11	CSTC1	R	WWSS		WWSH		RWSS		RWSH		0	0	0	0	0	MDAR		
		W																
$12	CSTC2	R	WWSS		WWSH		RWSS		RWSH		0	0	0	0	0	MDAR		
		W																
$13	CSTC3	R	WWSS		WWSH		RWSS		RWSH		0	0	0	0	0	MDAR		
		W																
$18	BCR	R	DRV	BMDAR			0	0	BWWS					BRWS				
		W																

R	0	Read as 0
W		Reserved

Figure 5-2. EMI Register Map Summary

5.6 Register Descriptions (EMI_BASE = $1FFE40)

5.6.1 Chip Select Base Address Registers 0–3 (CSBAR0–CSBAR3)

The CSBAR registers are defined in **Figure 5-3.** It determines the active address range of a given $\overline{CS}n$. The Block Size (BLKSZ) field determines the size of the memory map covered by the $\overline{CS}n$. This field also determines which of the address bits to use when specifying the base address of the $\overline{CS}n$. This encoding is detailed in **Table 5-2.** When the active bits match the address and the constraints specified in the CSOR are also met, the $\overline{CS}n$ is asserted.

The chip-select address compare logic uses only the most significant bits to match an address within a block. The value of the base address must be an integer multiple of the block size (i.e.,

the base address can be at block size boundaries only). For example, for a block size of 64k, the base address can be 64k, 128k, 192k, 256k, 320k, etc.

Note: The default reset value for $\overline{CS}n$ will enable a 32K block of external memory starting at address zero. This may be defined to be something else for a specific chip in which case the chip user manual will detail the specific reset value.

Base + $0 - $3	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	ADR23	ADR22	ADR21	ADR20	ADR19	ADR18	ADR17	ADR16	ADR15	ADR14	ADR13	ADR12	BLKSZ			
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

Figure 5-3. Chip Select Base Address Registers 0–3 (CSBAR0–CSBAR3)

See Programmer's Sheet on Appendix page B-9

Table 5-2. CSBAR Encoding of the BLKSZ Field

BLKSZ	Block Size	Address Lines Compared
0000	4K	X: ADR[23:12], P: ADR[20:12]
0001	8K	X: ADR[23:13], P: ADR[20:13]
0010	16K	X: ADR[23:14], P: ADR[20:14]
0011	32K	X: ADR[23:15], P: ADR[20:15]
0100	64K	X: ADR[23:16], P: ADR [20:16]
0101	128K	X: ADR[23:17], P: ADR[20:17]
0110	256K	X: ADR[23:18], P: ADR[20:18]
0111	512K	X: ADR[23:19], P: ADR [20:19]
1000	1M	X: ADR[23:20], P: ADR[20:20]
1001	2M	X: ADR[23:21]. All program address space decoded.
1010	4M	X: ADR[23:22]. No program address space decoded.
1011	8M	X: ADR[23:23]. No program address space decoded.
1100	16M	All data address space decoded. No program address space decoded.
1101	Reserved	No data address space decoded. No program address space decoded.
1110	Reserved	No data address space decoded. No program address space decoded.
1111	Reserved	No data address space decoded. No program address space decoded.

5.6.2 Chip Select Option Registers 0–3 (CSOR0–CSOR3)

A Chip Select Option Register is required for every chip select. This register specifies the mode of operation of the chip select and the timing requirements of the external memory.

Note: The $\overline{CS}n$ logic can be used to define external memory wait states even if the $\overline{CS}n$ pin is used as GPIO.

Note: The $\overline{CS}n$ output can be disabled by setting either the PS/DS, R/W or BYTE_EN fields to zero.

Base + $8 - $B	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read / Write	RWS					BYTE_EN		R/W		PS/DS		WWS				
Reset	1	0	1	1	1	0	0	0	0	0	0	1	0	1	1	1

Figure 5-4. Chip Select Option Registers 0–3 (CSOR0–CSOR3)

See Programmer's Sheets on Appendix page C- 10

5.6.2.1 Read Wait States (RWS)—Bits 15–11

The RWS field specifies the number of additional system clocks, 0-30 (31 is invalid) to delay for read access to the selected memory. The value of RWS should be set as indicated in **Section 5.7.1.**

5.6.2.2 Upper/Lower Byte Option (BYTE_EN)—Bits 10–9

This field specifies whether the memory is 16 bits wide or one byte wide. If the memory is byte wide, the option of upper or lower byte of a 16-bit word is selectable. **Table 5-3** provides the encoding of this field.

Table 5-3. CSOR Encoding BYTE_EN Values

Value	Meaning
00	Disable
01	Lower Byte Enable
10	Upper Byte Enable
11	Both Bytes Enable

5.6.2.3 Read/Write Enable (R/W)—Bits 8–7

This field determines the read/write capabilities of the associated memory as shown in **Table 5-4.**

Table 5-4. CSOR Encoding of Read/Write Values

Value	Meaning
00	Disable
01	Write-Only
10	Read-Only
11	Read / Write

5.6.2.4 Program/Data Space Select (PS/DS)—Bits 6–5

The mapping of a chip select to program and/or data space is shown in **Table 5-5.**

Table 5-5. CSOR Encoding of PS/DS Values

Value	Meaning	
	Flash_Security_Enable = 0	Flash_Security_Enable = 1
00	Disable	Disable
01	DS Only	DS Only
10	PS Only	Disable
11	Both PS and DS	DS Only

5.6.2.5 Write Wait States (WWS)—Bits 4–0

The WWS field specifies the number of additional system clocks 0-30 (31 is invalid) to delay for write access to the selected memory. The value of WWS should be set as indicated in **Section 5.7.2.**

5.6.3 Chip Select Timing Control Registers 0–3 (CSTC0–CSTC3)

A Chip Select Timing Control (CSTC) register is required for every chip select. This register specifies the detailed timing required for accessing devices in the selected memory map. At reset, these registers are configured for minimal timing in the external access waveforms. Therefore, these registers need only be adjusted if required by slower memory/peripheral devices.

Base + $10 - $13	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	WWSS		WWSH		RWSS		RWSH		0	0	0	0	0	MDAR		
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

Figure 5-5. Chip Select Timing Control Registers 0–3 (CSTC0–CSTC3)

See Programmer's Sheet on Appendix page B-12

5.6.3.1 Write Wait States Setup Delay (WWSS)—Bits 15–14

This field affects the write cycle timing diagram, illustrated in **Figure 5-16.** Additional time (clock cycles) is provided between the assertion of $\overline{CSn}$ and address lines and the assertion of $\overline{WR}$. The value of WWSS should be set as indicated in **Section 5.7.2.**

5.6.3.2 Write Wait States Hold Delay (WWSH)—Bits 13–12

This field affects the write cycle timing diagram, illustrated in **Figure 5-17.** The WWSH field specifies the number of additional system clocks to hold the address, data, and $\overline{CSn}$ signals after the $\overline{WR}$ signal is deasserted. The value of WWSH should be set as indicated in **Section 5.7.2.**

5.6.3.3 Read Wait States Setup Delay (RWSS)—Bits 11–10

This field affects the read cycle timing diagram, illustrated in **Figure 5-10.** Additional time (clock cycles) is provided between the assertion of $\overline{\text{CS}n}$ and address lines and the assertion of $\overline{\text{RD}}$. The value of RWSS should be set as indicated in **Read Timing.**

5.6.3.4 Read Wait States Hold Delay (RWSH)—Bits 9–8

This field affects the read cycle timing diagram, illustrated in **Figure 5-11.** The RWSH field specifies the number of additional system clocks to hold the address, data, and $\overline{\text{CS}n}$ signals after the $\overline{\text{RD}}$ signal is deasserted. The value of RWSH should be set as indicated in **Section 5.7.1.**

Note: If both, the RWSS and RWSH fields are set to zero the EMI read timing is set for consecutive mode. In this mode the $\overline{\text{RD}}$ signal will remain active during back-to-back reads from the same $\overline{\text{CS}n}$ controlled memory space.

5.6.3.5 Reserved—Bits 7–3

This bit field is reserved or not implemented. It is read as 0 and cannot be modified by writing.

5.6.3.6 Minimal Delay After Read (MDAR)—Bits 2–0

This field specifies the number of system clocks to delay between reading from memory in a $\overline{\text{CS}n}$ controlled space and reading from another device. Since a write to the device implies activating the digital signal controller on the bus, this is also considered a read from another device.

Figure 5-6 illustrates the timing issue requiring the introduction of the MDAR field. In this diagram, $\overline{\text{CS1}}$ is assumed to operate a slow flash memory in P-space while $\overline{\text{CS2}}$ is operating a faster RAM in X-space. In some bus contention cases, it is possible to encounter data integrity problems where the contention is occurring at the time the data bus is sampled.

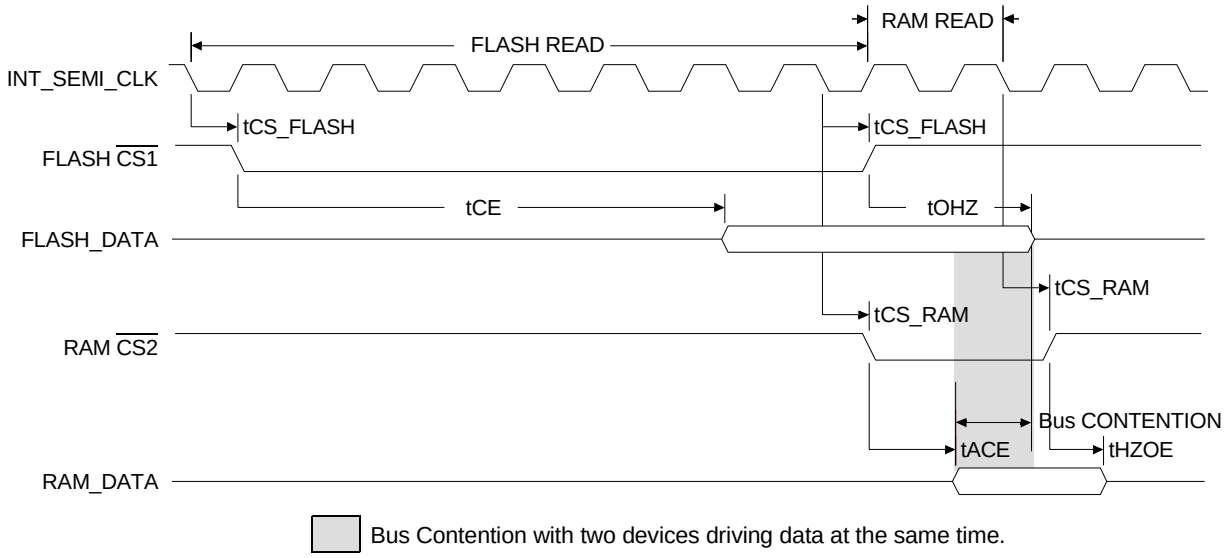


Figure 5-6. Data Bus Contention Timing Requiring MDAR Field Assertion

5.6.4 Bus Control Register (BCR)

The BCR register defines the default read timing for external memory accesses to addresses not covered by the $\overline{CS}$/CSOR/CSTC registers. The timing specified by the BCR register applies to both program and data space accesses because the PS and DS control signals are not directly available on the chip pinouts.

Note: Any of the $\overline{CSn}$ signals can be configured to mirror the $\overline{PS}$ and/or $\overline{DS}$ function, but then the associated $\overline{CSn}$ configuration registers control the timing.

Base + $18	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	DRV	BMDAR			0	0	BWWS					BRWS				
Write																
Reset	0	0	0	0	0	0	1	0	1	1	1	1	0	1	1	1

Figure 5-7. Bus Control Register (BCR)

See Programmer's Sheets on Appendix page C- 12

5.6.4.1 Drive (DRV)—Bit 15

The Drive (DRV) control bit is used to specify what occurs on the external memory port pins when no external access is performed (whether the pins remain driven or are placed in tri-state). **Table 5-6** summarizes the action of the EMI when the DRV bit is cleared or is set. DRV bit is cleared on hardware reset, but should be set in most customer applications.

Table 5-6. Operation with DRV

800E Core Operating State	DRV	Pins		
		A23:A0	$\overline{RD}$, $\overline{WR}$, $\overline{CSn}$	D15:D0
EMI is Between External Memory Accesses	0	Tri-stated	Tri-stated	Tri-stated
Reset Mode		Tri-stated	Pulled High Internally	Tri-stated
EMI is Between External Memory Accesses	1	Driven	Driven ($\overline{RD}$, $\overline{WR}$, $\overline{CSn}$ are Deasserted)	Tri-stated
Reset Mode		Tri-stated	Pulled High Internally	Tri-stated

5.6.4.2 Base Minimal Delay After Read (BMDAR)—Bits 14–12

This bit field specifies the number of system clocks to delay after reading from memory *not* in $\overline{CS}$ controlled space. Since a write to the device implies activating the digital signal controller on the bus, this is also considered a read from another device, therefore activating the BMDAR timing control. Please see the description of the MDAR field of the CSTC registers for a discussion of the function of this control.

5.6.4.3 Reserved—Bits 11–10

This bit field is reserved or not implemented. It is read as 0 and cannot be modified by writing.

5.6.4.4 Base Write Wait States (BWWS)—Bits 9–5

This bit field specifies the number of additional system clocks 0-30 (31 is invalid) to delay for write access to the selected memory when the memory address does *not* fall within $\overline{CS}$ controlled range. The value of BWWS should be set as indicated in **Timing Specifications.**

5.6.4.5 Base Read Wait States (BRWS)—Bits 4–0

This bit field specifies the number of additional system clocks 0-30 (31 is invalid) to delay for read access to the selected memory when the memory address does *not* fall within $\overline{CS}$ controlled range. The value of BRWS should be set as indicated in **Section 5.7.**

5.7 Timing Specifications

5.7.1 Read Timing

5.7.1.1 Consecutive Mode Operation

Figure 5-8 illustrates the read timing for external memory access**.** For comparison, a single read cycle is illustrated followed by a null cycle and then a back-to-back read.

Figure 5-8 assumes zero wait states are required for the access. **Figure 5-9** illustrates a timing diagram with one wait state added.

There are two read setup timing parameters for each read cycle. The core will latch the data on the rising edge of the internal clock while t_{RSD} indicates the core setup time. The external timing of the address and controls is adjusted so they may be changing at this time. Therefore, a data latch is introduced to capture the data (at the pin) a quarter clock earlier, on the rising edge of the internal delayed clock. The setup time required for this latch is illustrated by t_{RSDP} in the diagrams. For slow clock speeds, t_{RSDP} is more critical, while t_{RSD} may be harder to meet for faster clock rates.

Note: During back-to-back reads, $\overline{RD}$ remains low to provide the fastest read cycle time.

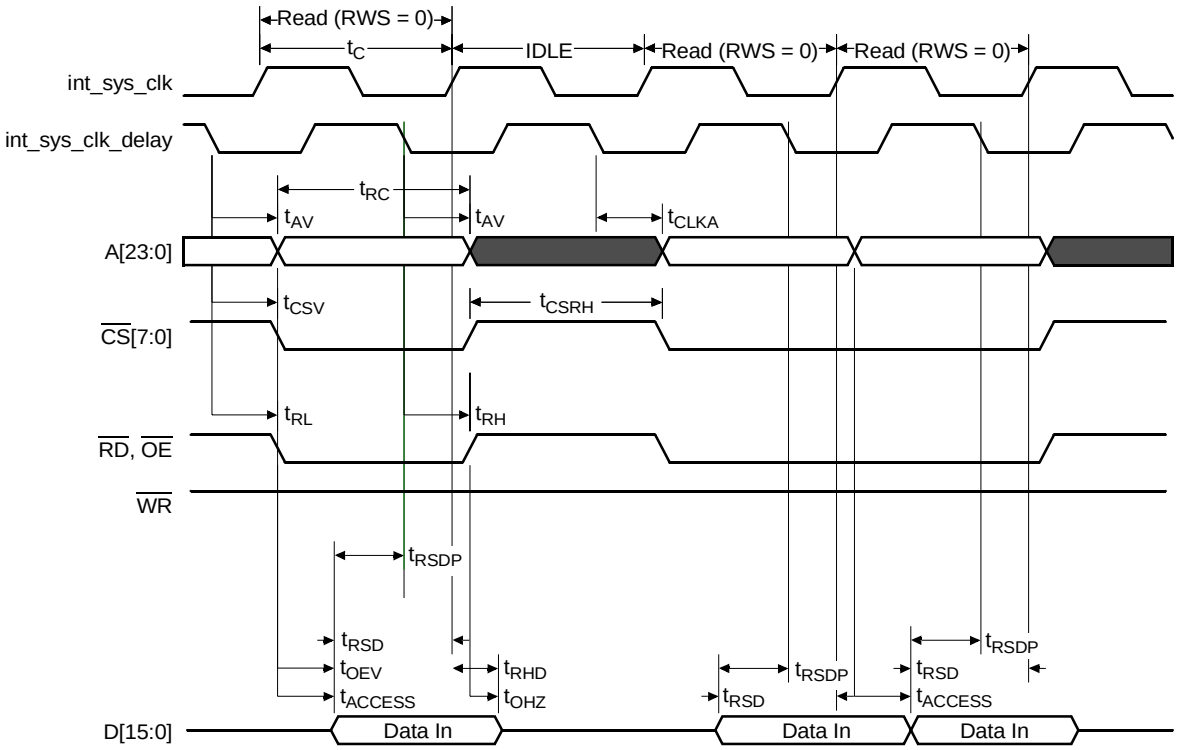


Figure 5-8. External Read Cycle with Clock and RWS = 0

Note: INT_SYS_CLK is the internal system clock from which everything is referenced.

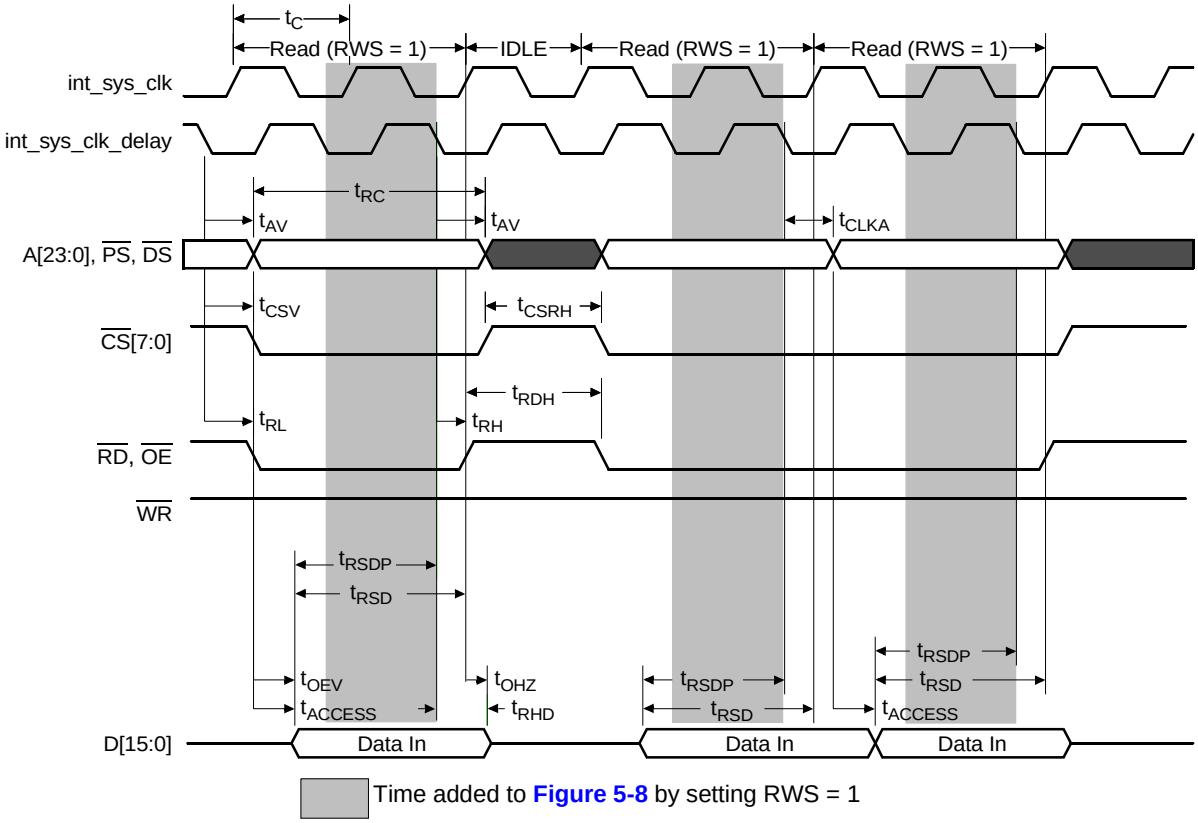


Figure 5-9. External Read Cycle with RWS = 1, RWSH = 0 and RWSS = 0

5.7.1.2 Read Setup and Hold Timing

Although most memory devices can perform consecutive reads by holding the $\overline{CSn}$ and $\overline{RD}(\overline{OE})$ signals in the active state and changing the address, there are peripheral devices that require $\overline{RD}(\overline{OE})$ to transition to the inactive state between reads of certain registers. This timing can be accommodated with the Read Setup (RWSS) and/or Read Hold (RWSH) control fields illustrated in **Figure 5-10** and **Figure 5-11.**

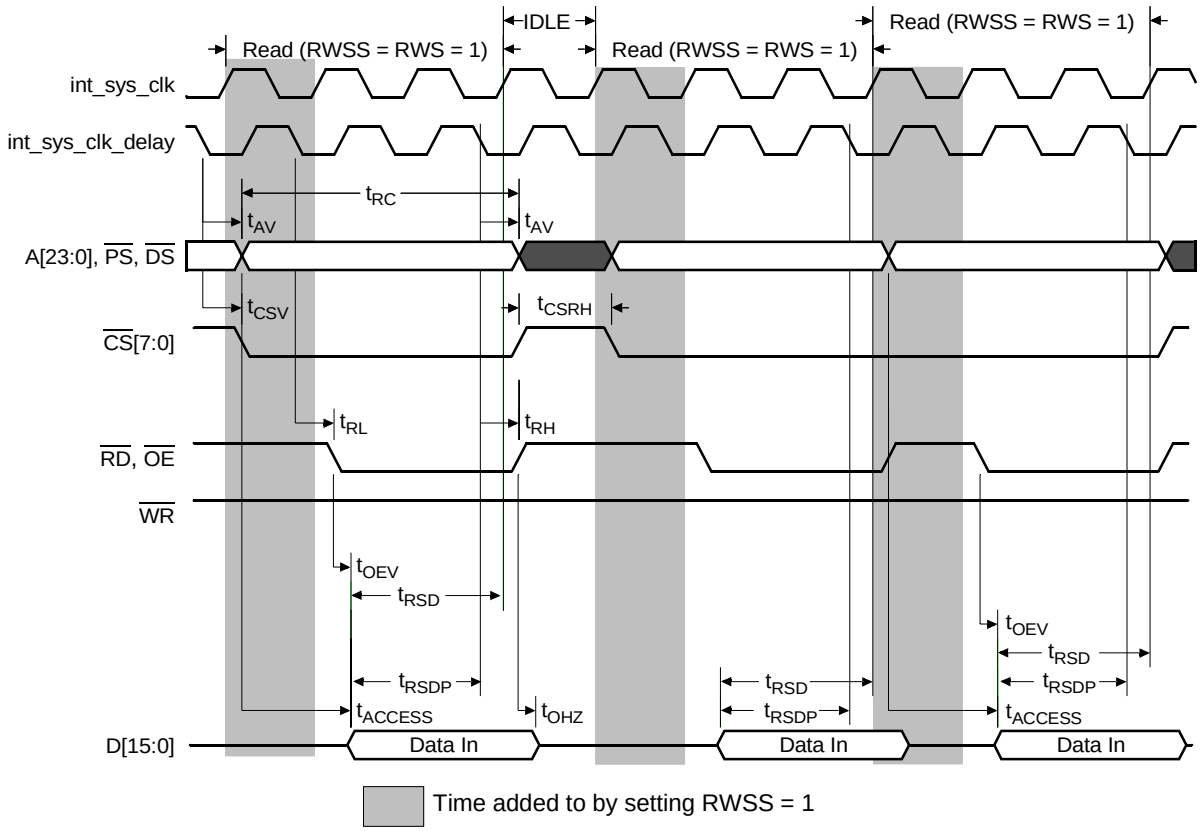


Figure 5-10. External Read Cycle with RWSS = RWS = 1, and RWSH = 0

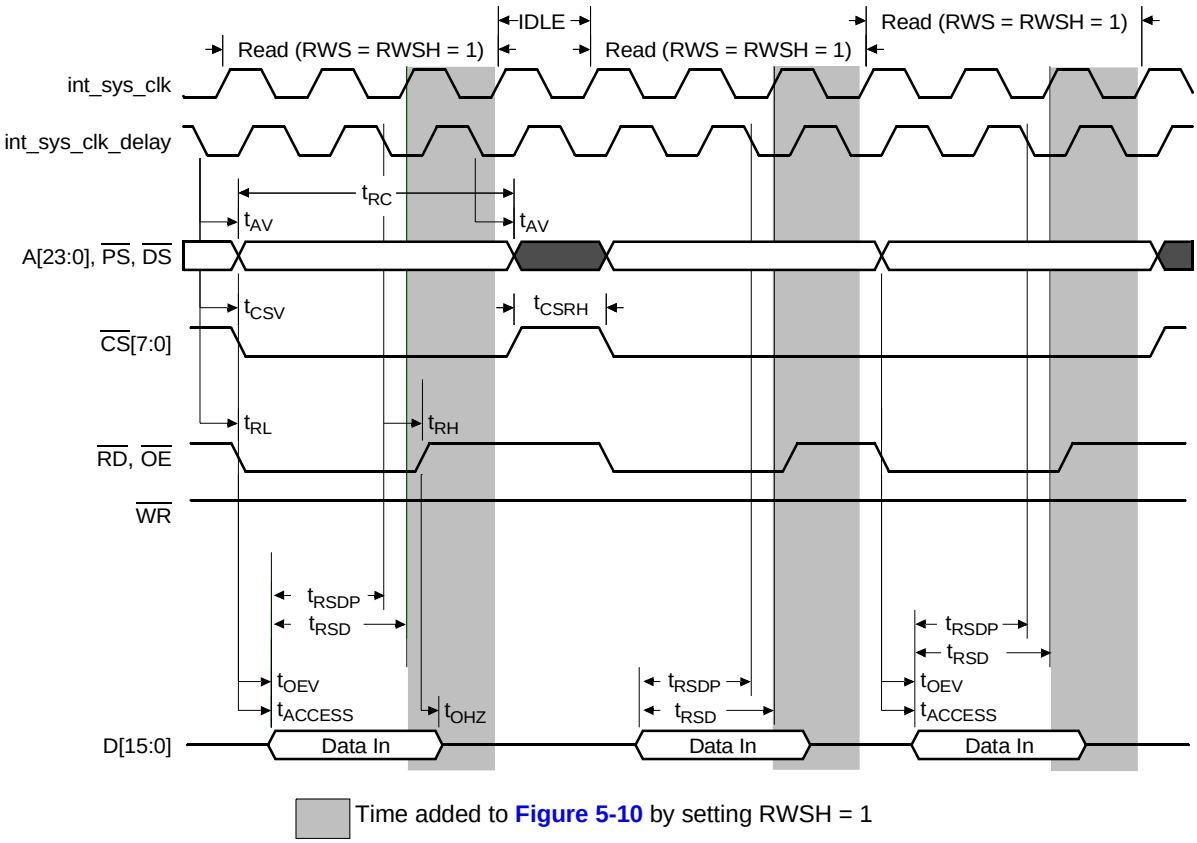


Figure 5-11. External Read Cycle RWS = RWSH = 1 and RWSS = 0

5.7.2 Write Timing

Figure 5-12 shows the write timing for external memory access**.** For comparison, a single write cycle is shown followed by a null cycle and then a back-to-back write.This figure assumes zero wait states are required for the access.

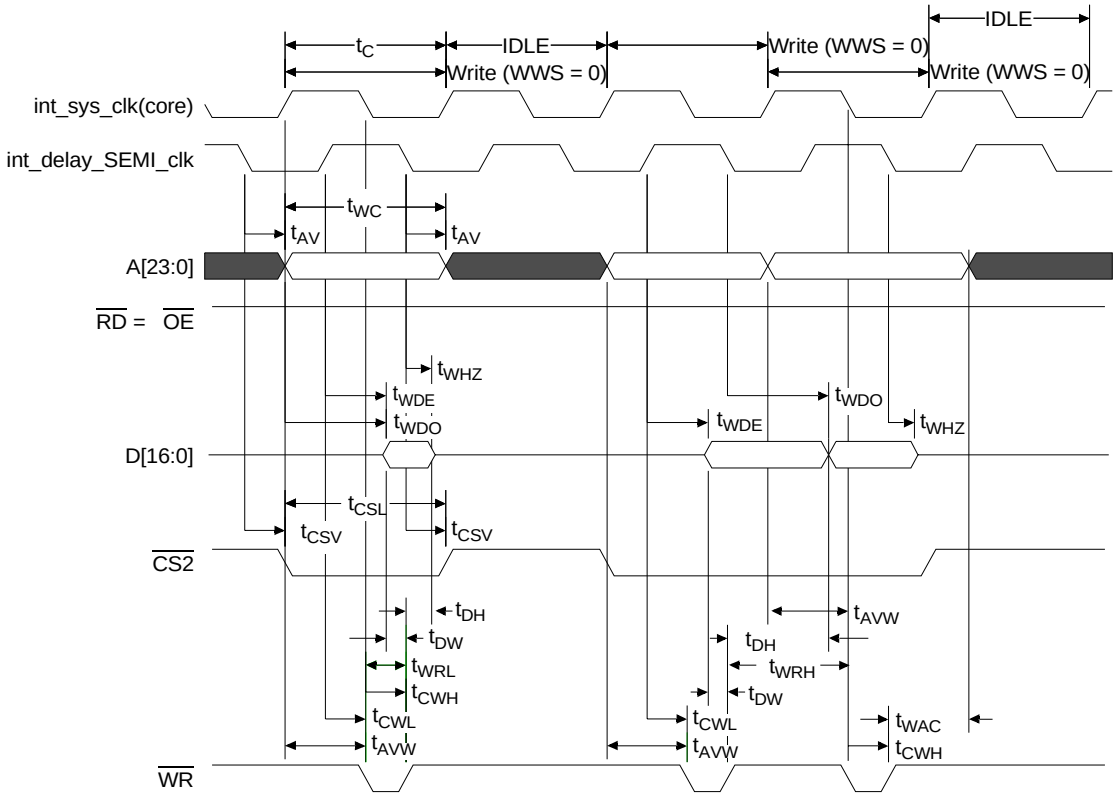


Figure 5-12. External Write Cycle

Note: When WWS = 0 the timing of the $\overline{WR}$ strobe is generated from different clock edges than when it is set to some other value. This change in timing allows the possibility of single cycle write operation, but reduces the pulse width of $\overline{WR}$ to one quarter clock. This may make it difficult to meet write timing requirements for most devices when operating at normal clock rates.

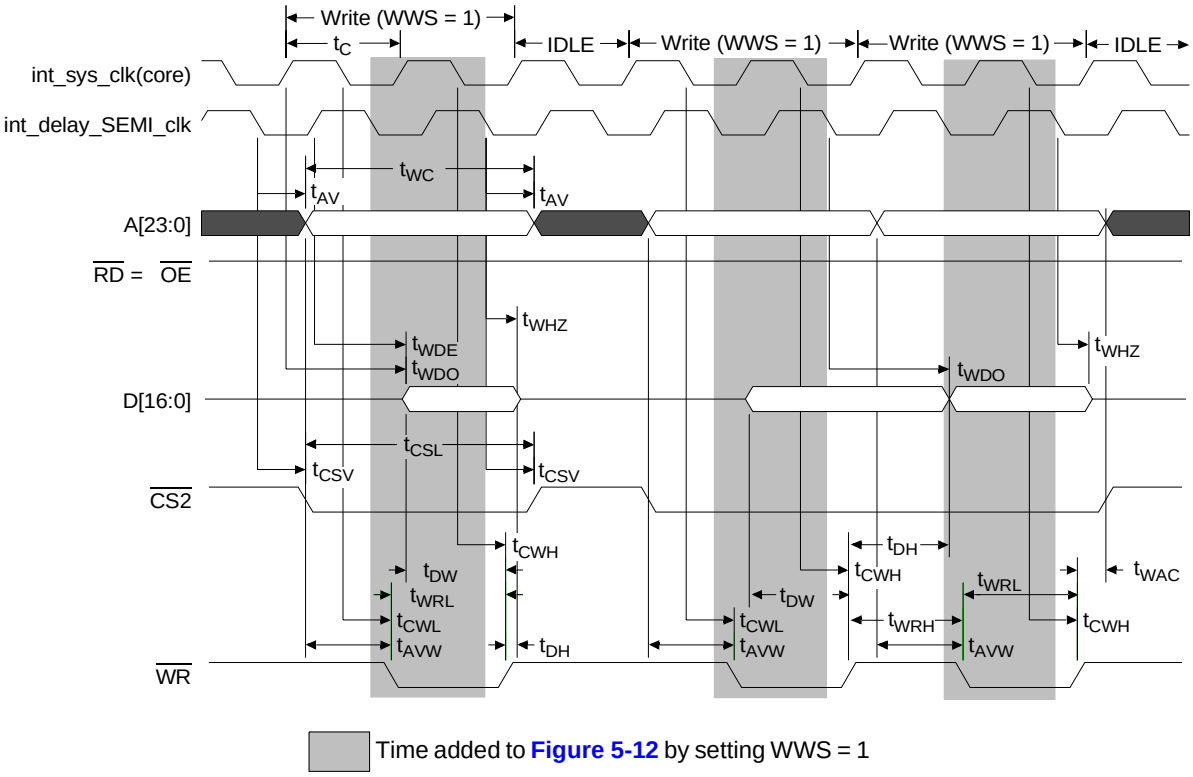


Figure 5-13. External Write Cycle with WWS = 1, WWSH = 0, and WWSS = 0

5.7.2.1 Write Setup and Hold Timing

Since the timing of the strobes is different when WWS = 0 than it is when WWS > 0, two sets of timing diagrams are illustrated in **Figure 5-14, Figure 5-15, Figure 5-16,** and **Figure 5-17.**

5.7.2.2 WWS = 0

Although most memory devices require a zero setup and hold time, there are some peripheral devices where a setup/hold time is required. The WWSS and WWSH field of the CSTC register provides the ability to allow for a write setup and/or hold time requirement as shown in **Figure 5-14** and **Figure 5-15.**

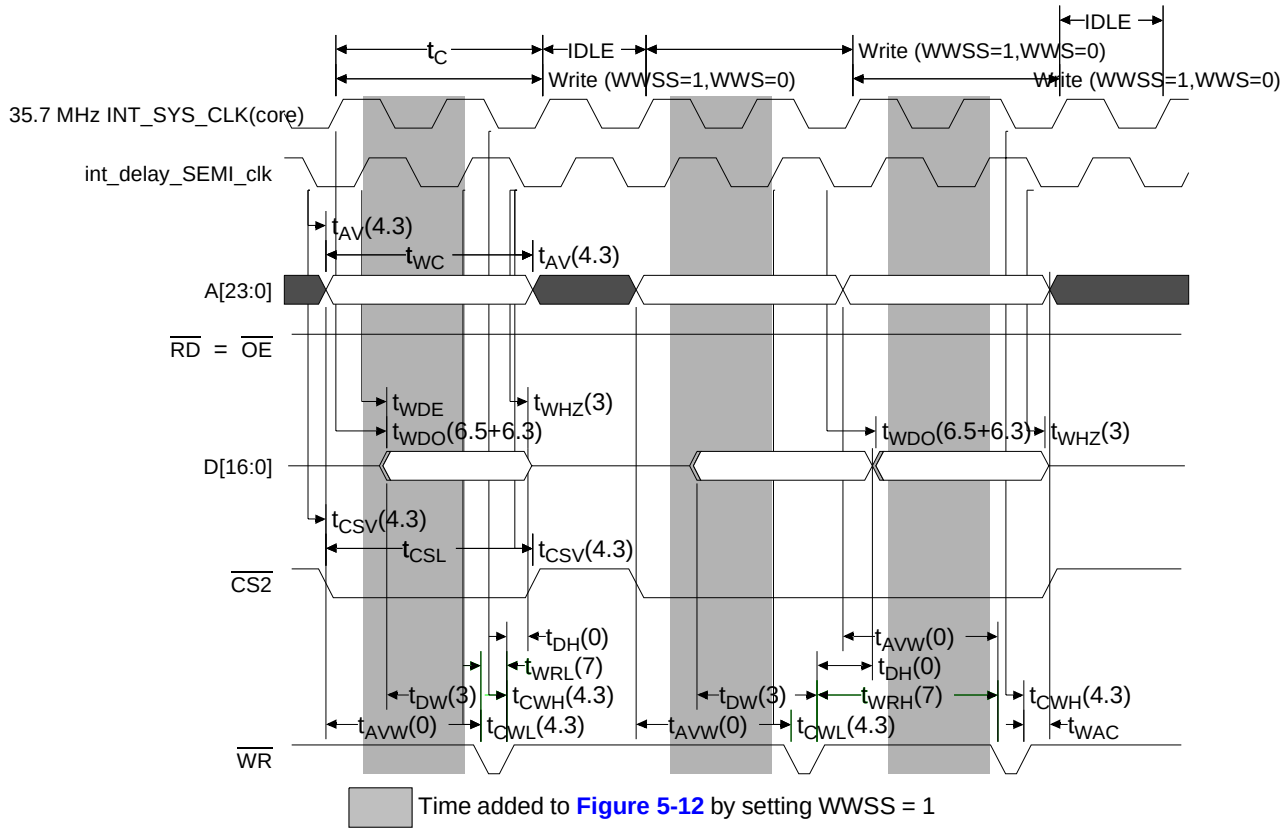


Figure 5-14. External Write Cycle with WWSS = 1, WWS = 0 and WWSH = 0

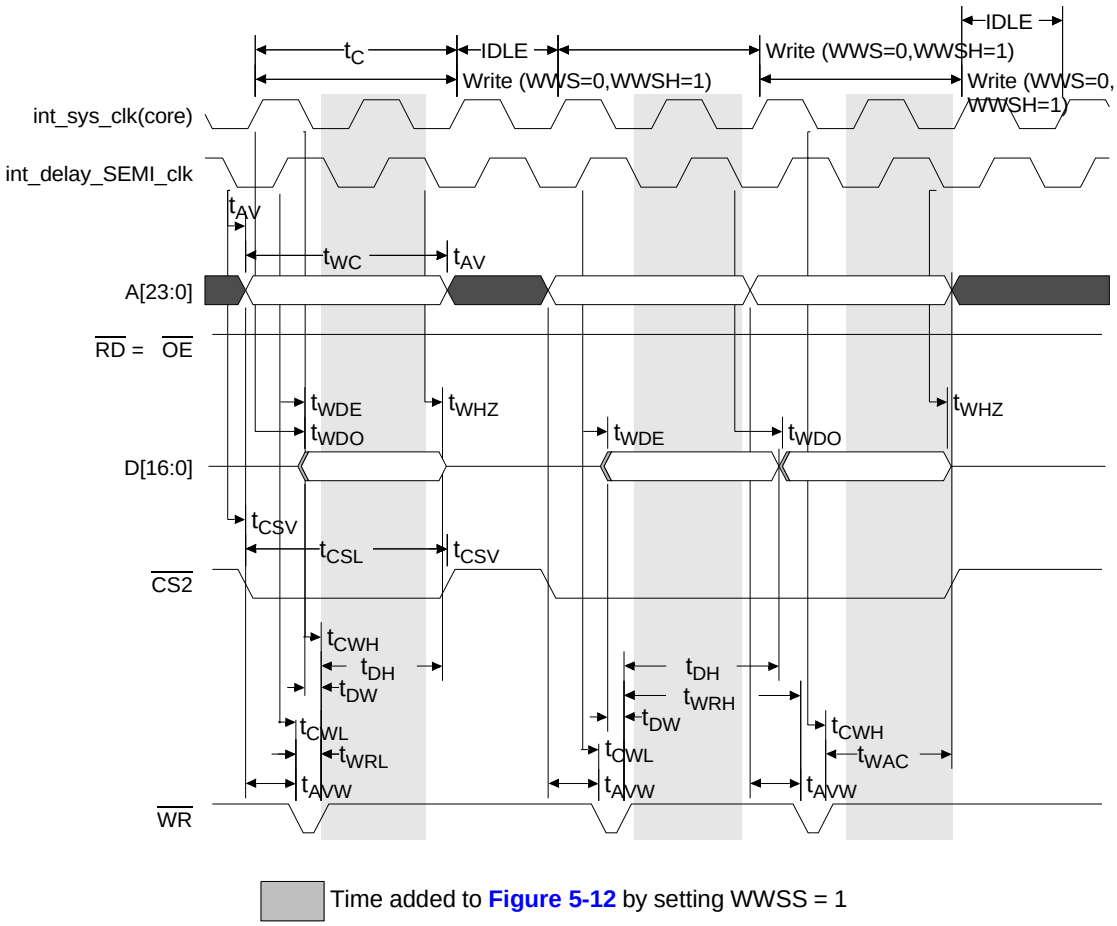


Figure 5-15. External Write Cycle with WWS = 0, WWSH = 1, WWSS = 0

5.7.2.3 WWS > 0

Although most memory devices require a zero setup and hold time, there are some peripheral devices where a setup/hold time is required. The WWSS and WWSH field of the CSTC register provides the ability to allow for a write setup and/or hold time requirement as shown in **Figure 5-16** and **Figure 5-17** respectively.

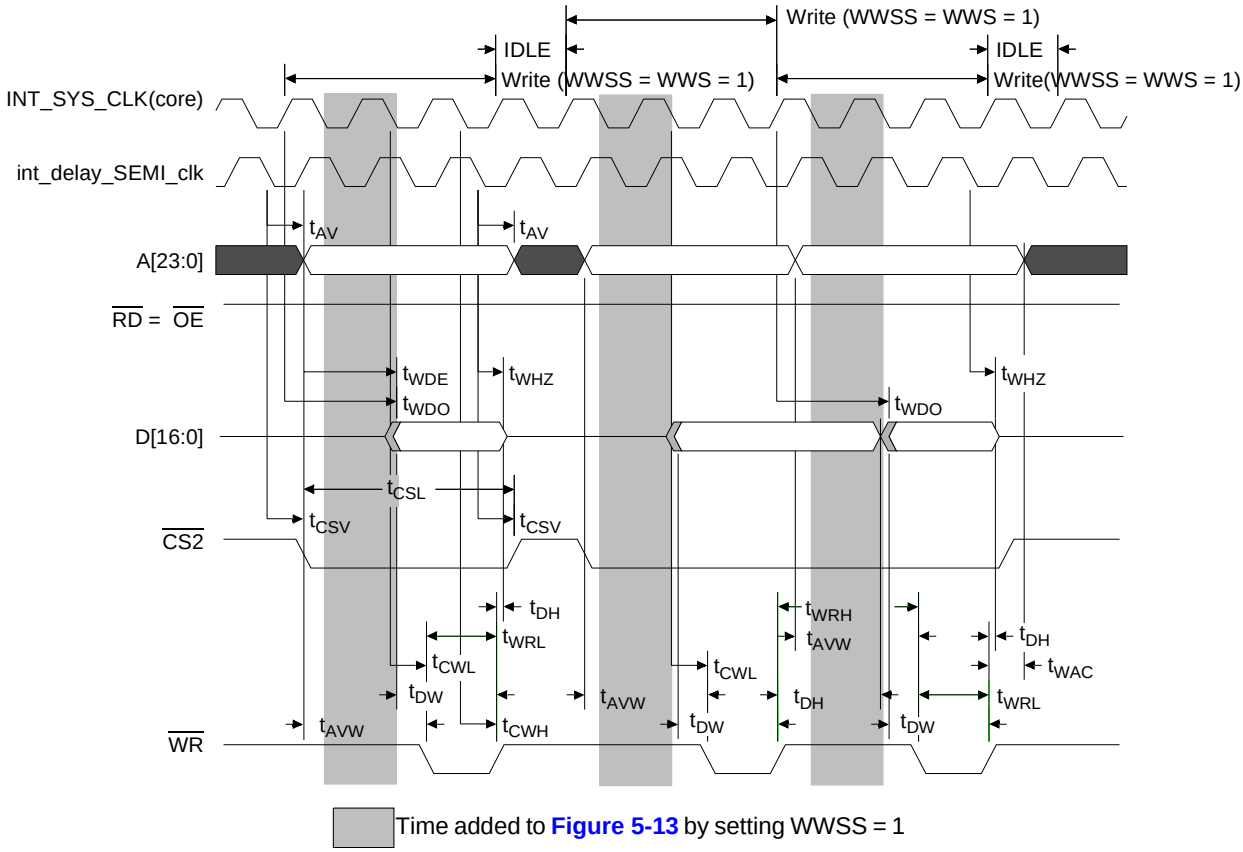


Figure 5-16. External Write Cycle with WWSS = WWS = 1 and WWSH = 0

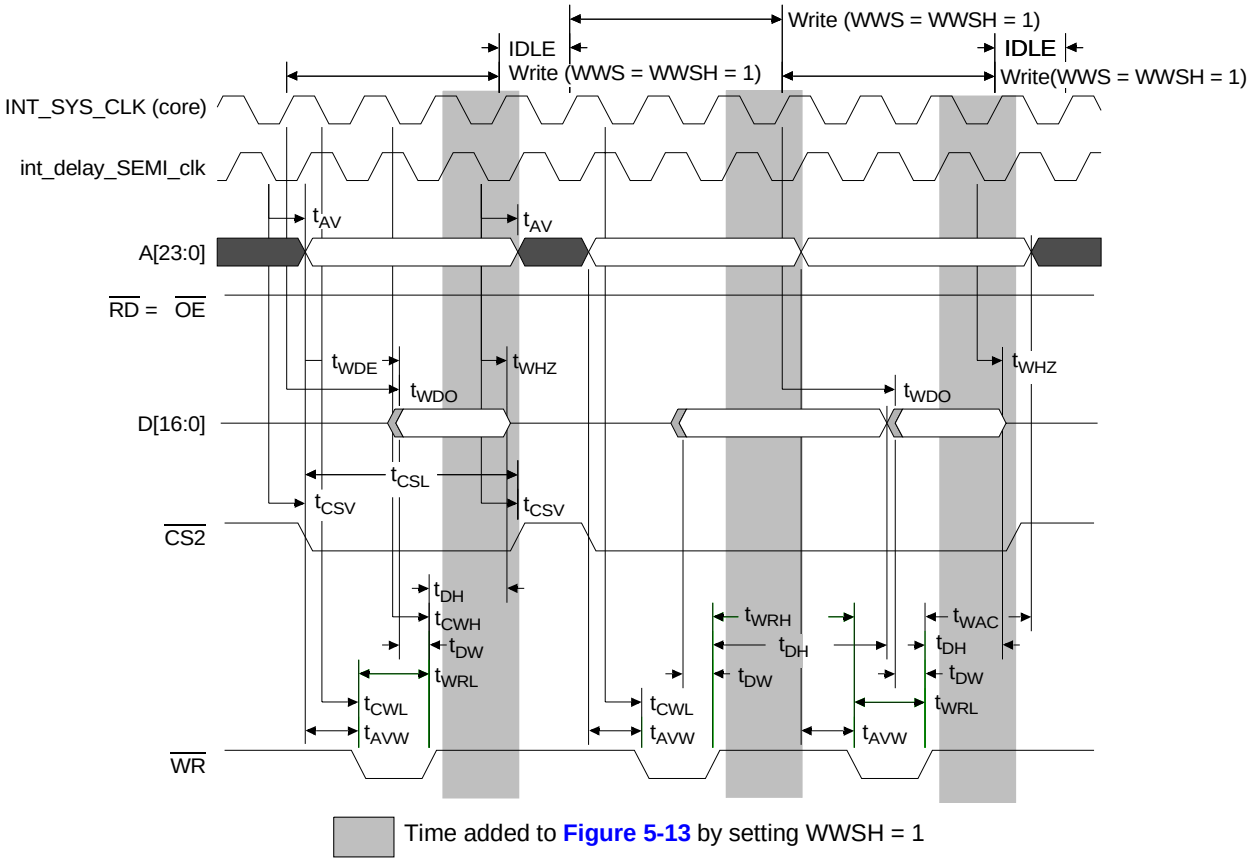


Figure 5-17. External Write Cycle with WWS = WWSH = 1 (WWSS = 0)

5.1 Clocks

The EMI operates from clocks internal to the chip and does not require/provide clocks external to the chip.

5.1 Interrupts

There are no interrupts generated by this module.

5.1 Resets

All reset types are equivalent for the EMI and therefore have the same effect. The EMI outputs during reset are controlled by the DRV bit of the BCR. During reset this bit is set to zero. Therefore, **Table 5-6** defines the reset state of all EMI pins.

Chapter 6
On-Chip Clock Synthesis (OCCS)

6.1 Introduction

The On-Chip Clock Synthesis (OCCS) module allows product design using an inexpensive 4MHz crystal or an external clock source to run the DSP56853/54/55/57/58 at any frequency from 0 to 120MHz. The OCCS module is comprised of two major blocks: the Oscillator (OSC), and the PLL/CGM (analog - Phase Locked Loop/digital - Clock Generation Module (CGM)). The OSC output clocks feed the PLL/CGM block. The PLL/CGM generates a time clock for Computer Operating Properly (COP) timer use. The PLL/CGM also generates a master clock consumed by the System Integration Module (SIM). The SIM generates derivative clocks for consumption by the core logic and IPBus peripherals.

The SIM divides the MSTR_CLK (typically 240MHz) by 2 to create the 56800E core clock (typically 120MHz) and by 4 to create the IPBUS_CLK (typically 60 MHz). All peripherals on the DSP56853/54/55/57/58 run off the IPBus clock frequency. The COP and TOD peripherals also consumes the much lower frequency TIME_CLK (typically 31.25 KHz).

The PLL may be used to generate a high frequency clock from the low-frequency crystal-referenced (or external clock driven) OSC circuit. The PLL provides an exact integer multiple of the oscillator's output reference frequency (Fref). The frequency multiplication is in the range of 20 to 120.

The CGM controls the PLL's output frequency. The CGM also selects between the PLL (PLL_OUT) and OSC (Fref) as potential master clock sources and routes the selection to the SIM. The CGM also contains circuitry to detect if the PLL is unlocked and generates an interrupt signal for the condition.

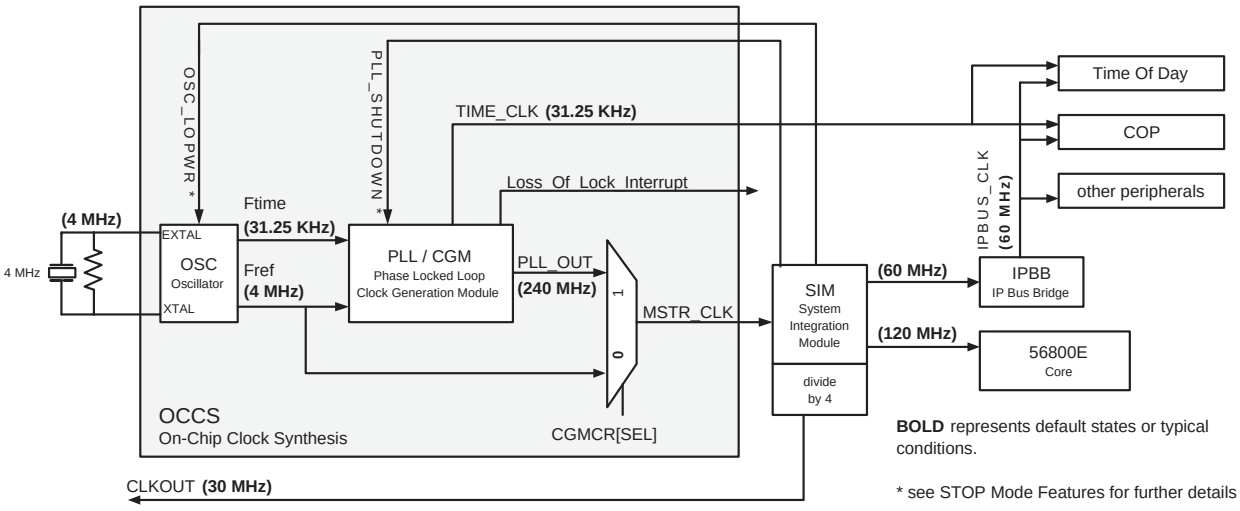


Figure 6-1. OCCS Integration Overview

Out of reset, the CGMCR[SEL] control bit is 0, selecting the Fref path as the source of MSTR_CLK. The core will proceed to execute code using a clock divided down from the

oscillator's Fref output. The core will run at Fref/2 and the IPBUS_CLK will run at Fref/4. Among the first things applications typically do is turn on the PLL, wait for lock indication and set CGMCR[SEL] to 1, enabling high speed operation.

6.1.1 OCCS Features

- OSC connects to external crystals in the range of 2 to 4MHz
- OSC can optionally accept an external active clock (0 to 240MHz)
- PLL generates any integer multiple frequency, allowing DC to 120MHZ execution
- CGM provides glitch-free transition between OSC and PLL clock sources
- CGM provides digital loss of lock detection
- Ultra Low Power modes are available while COP timer and TOD are kept alive

6.2 OSC (Oscillator) Circuit Detail

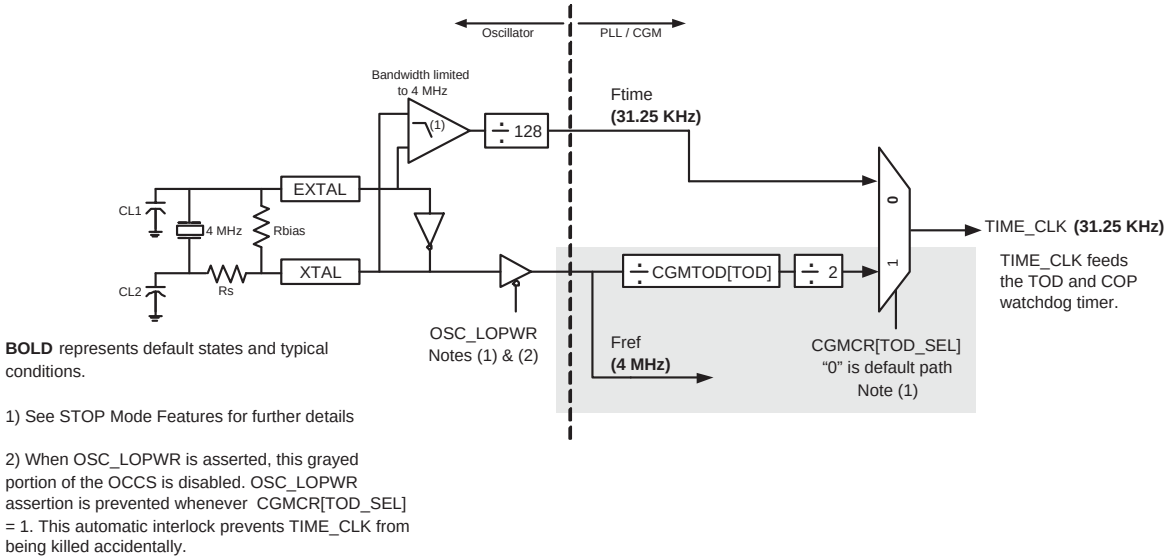


Figure 6-2. OSC Supplying Clocks to PLL/CGM

A typical connection for the OSC module is shown above. The weak differential signal coming in directly from EXTAL/XTAL pin pair is routed to a very low power differential amplifier. Because the amplifier is so low in power, it has a usable bandwidth limit somewhat above just 4MHz. The resulting clock frequency is divided down by a fixed value of 128 to yield a 31.25KHz clock. Out of reset, the register CGMCR[TOD_SEL] is 0, so this is the path selected through the mux to create the TIME_CLK used by the COP and TOD. If the signal present on XTAL is above 4MHz (e.g. driven by an external active clock) then it is necessary to set CGMCR[TOD_SEL] to 1.

The signal from XTAL is buffered up (becoming Fref) and is consumed everywhere else.

6.2.1 Using an External Crystal

Figure 6-3 illustrates the typical application details for using an external crystal. A 4MHz, At cut, parallel resonant crystal is mounted across XTAL and EXTAL pins. A 1 Mohm bias resistor is paralleled to that connection. The default path for TIME_CLK generation (the differential amplifier path) is recommended and therefore CGMCR[TOD] register setting is a don't care. Please see **Section 6.2.4** for further details.

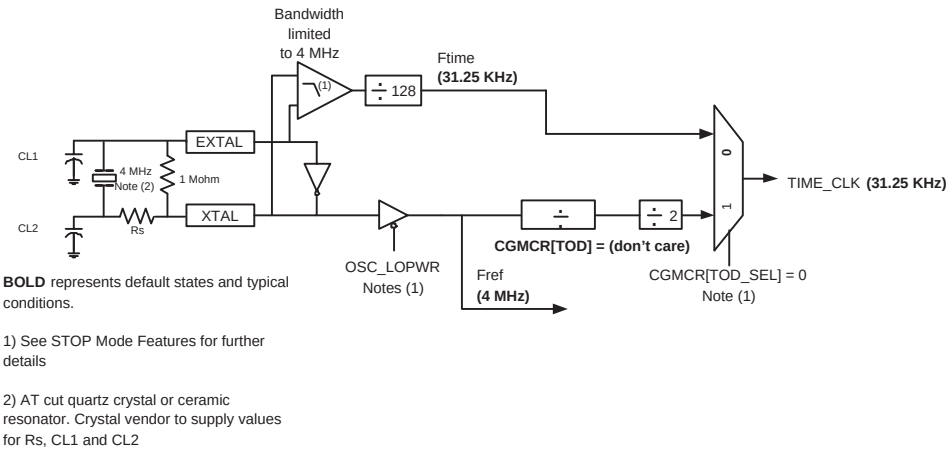


Figure 6-3. Using an External Crystal

The values of Rs, CL1 and CL2 are determined with assistance of your crystal manufacturer. In general, CL1 and CL2 are used to pull the crystal to the intended frequency and establish the Equivalent Series Resistance (ESR). Rs eliminates some of the inverter's gain (by an amount appropriate for the resulting ESR).

Note: The CGMCR register's TOD_SEL field may be set to either 0 or 1. The recommended setting of 0 allows very low power operation when executing a STOP instruction.

6.2.2 Using an External Active Clock Source Below 4 MHz

When using an external active clock source of a frequency less than, or equal to, 4MHz then the connections shown below are recommended. Here, the EXTAL pin is biased to 1.65V and XTAL is driven with a 0 to 3.3V square wave clock signal. The TIME_CLK source path is the same as above, using the fixed divide by 128 block and channel 0 of the MUX.

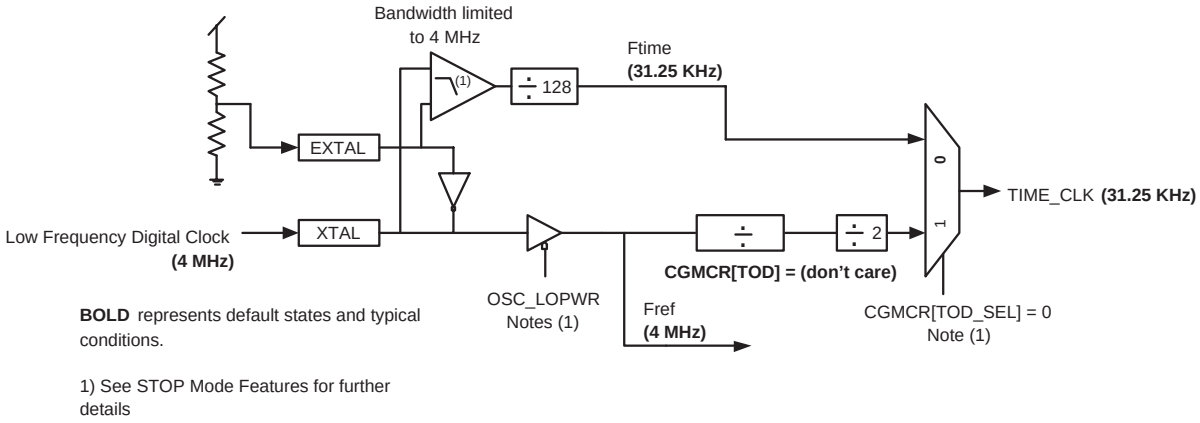


Figure 6-4. Using an External Active Low Frequency Clock, < 4 MHz

Note: The CGMCR register's TOD_SEL field may be set to either 0 or 1. The recommended setting of 0 allows very low power operation when executing a STOP instruction. If, however, a setting of 1 is used, then EXTAL can be tied to ground, to mid-rail (as shown), or high. If TOD_SEL is set to 1, then the optimal connection of EXTAL is to ground.

6.2.3 Using an External Active Clock Source Above 4 MHz

When using an external active clock source of a higher frequency than 4MHz (up to 240MHz is allowable) the settings detailed below should be used.

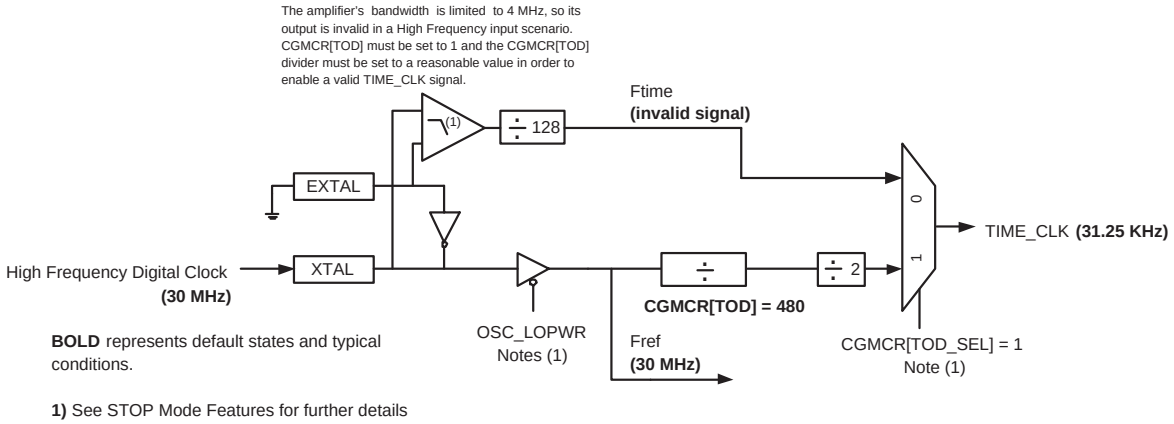


Figure 6-5. Using an External Active High Frequency Clock, > 4 MHz

Since the differential amplifier is band limited to just over 4MHz, in this example the default TIME_CLK path's divide by 128 counter will no longer have an adequate signal for proper operation. Instead, the user programmable divide by circuit should be used. This requires correct setting of both the CGMCR[TOD] and [TOD_SEL] register fields as shown in the figure.

In this particular example, the externally applied clock, and hence Fref, are running at 30 MHz. This requires TOD_SEL be set to 1 and the TOD register set to 480. The input frequency of 30MHz divided down by 480 and then again by a fixed value of 2 yields the desired 31.25KHz TIME_CLK frequency.

Note: With the CGMCR register's TOD_SEL field set to 1, EXTAL can be tied to ground (as shown), to mid-rail, or high. The optimal connection of EXTAL is to ground.

6.2.4 STOP Mode Features

In an attempt to conserve power, applications may power the device down by executing STOP or WAIT instructions. The DSP5685x OCCS module supports three variants of STOP mode processing.

1. Case where CGMCR[TOD_SEL] = 0

   A STOP instruction will result in the MSTR_CLK clock source being forced back to Fref and the PLL being powered down (PLL_SHUTDOWN asserts. If CGMCR[TOD_SEL] is 0, then OSC_LOWPWR will assert, bringing the OSC module into its lowest power alive state. Only the inverter, differential amplifier and the fixed divide-by 128 block remain

enabled, but that is enough to keep TIME_CLK and the associated COP counter alive and running. Fref is held quiescent.

When waking up from this *deep stop*, the PLL will need to be re-started, lock status re-queried and the PLL output re-selected to source MSTR_CLK.

2. Case where CGMCR[TOD_SEL] =1

   A STOP instruction will result in the MSTR_CLK clock source being forced back to Fref and the PLL being powered down (PLL_SHUTDOWN asserts). If CGMCR[TOD_SEL] is 1 when the STOP instruction is executed, then the SIM *will not* assert OSC_LOWPWR for in doing so, the TIME_CLK (and COP) would be killed. This is usually an undesirable situation from an applications perspective.

3. Fast STOP Recovery

   A *Fast STOP Recovery* is available in which *neither* the OSC_LOWPWR or PLL_SHUTDOWN signal are asserted. As such, no time is required to re-start the PLL, but it comes at the cost of increased power consumption by the PLL during STOP. Fast Stop Recovery is available by setting the OMR register's bit 6 to 1.

For further details on STOP and Fast STOP Recovery, please see **Section 4.9,** Power Mode Controls.

6.3 PLL (Phase Locked Loop) Circuit Detail

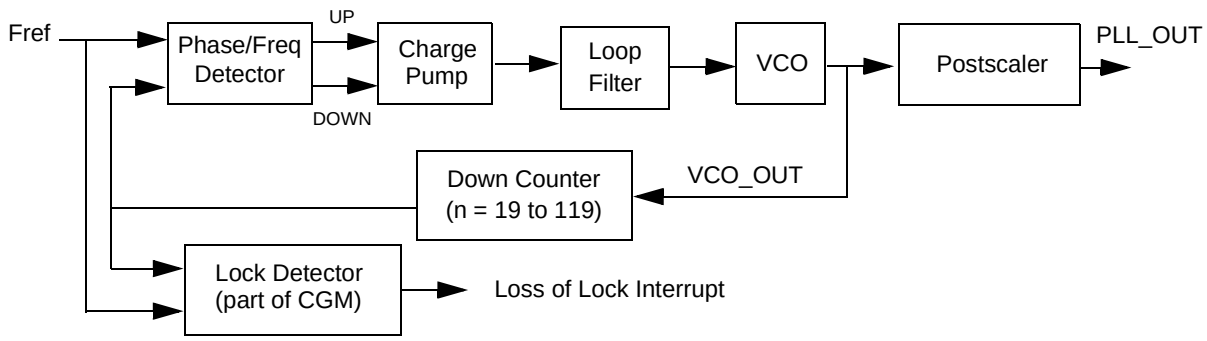


Figure 6-6. PLL Block Diagram

6.3.1 Phase Frequency Detector

The Phase Frequency Detector (PFD) compares the clock signal from the feedback divider to the input clock. When the input clock comes before the feedback clock the PFD generates a *down* pulse signal. The down pulse signal continues until the feedback clock signal arrives. If the feedback clock arrives at the PFD before the input clock, the PFD generates an *up* pulse, continuing until the input clock signal arrives.

6.3.2 Charge Pump

The Charge Pump draws charge into or out of the Loop Filter depending upon the signals from the Phase Frequency Detector. As charge is added to the Loop Filter the VCO control voltage increases. As charge is pulled out of the Loop Filter, the VCO control voltage decreases.

6.3.3 Loop Filter

The Loop Filter produces a voltage proportional to the amount of charge pumped into or out of the Loop Filter by the charge pump. The Loop Filter is a single pole RC filter.

6.3.4 Voltage Controlled Oscillator

The Voltage Controlled Oscillator (VCO) produces a frequency inversely proportional to the value of the control voltage signal coming out of the Loop Filter. The VCO gain is approximately 109MHz/Volt. The VCO has a frequency range of 80MHz to 380MHz with a center frequency of 240MHz.

6.3.5 Down Counter

The Down Counter is a programmable divide by *n* counter where the divide integer *n* is user-set to develop the PLL output frequency of interest. By presenting only one return pulse out of *n* input pulses to the return clock of the phase frequency detector, the PFD will drive the charge pump to raise the VCO frequency until the Down Counter return signal is in frequency and phase lock with the input clock signal. The output of the VCO will, therefore, be *n* times the frequency of the input clock signal. The value of *n* has a valid range of 19 to 119. The selected value of *n* depends upon the desired VCO output frequency and the input clock frequency (Fref). For the 2MHz input crystal the valid range of *n* will range from 39 to 119, producing a VCO frequency output of 80MHz to 240MHz according to the formula:

$$Fvco\_out = Fref \times (n+1)$$

The VCO's output frequency is routed through a postscaler so the final PLL output frequency is given by:

$$Fpll\_out = Fvco\_out / 2^{m}$$

where *m* is the value on the postscaler and can range from 0 to 7.

Example: Let Fref = 32MHz, n = 4, and m = 3, using the formulas gives:

$$Fpll\_out = (32MHz \times (4+1)) / 8$$

$$Fpll\_out = 20MHz.$$

For the 4MHz input crystal the valid range of *n* will be from 19 to 59, producing the same VCO output range, 80MHz to 240MHz.

6.3.6 PLL Lock Time User Notes

The PLL's Voltage Controlled Oscillator (VCO) has a characterized operating range extending from 80MHz to 240MHz. The PLL is programmable via a divide by *n+1* register, able to take on values varying between 1 and 128. For higher values of *n*, PLL lock time becomes an issue. It is recommended to avoid values of *n* resulting in the VCO frequency being greater than 240MHz. The graphic in **Figure 6-7** depicts the range of recommended output frequencies of VCO_OUT, plotting *n versus* the input frequency (Fref). The lower the value of *n*, the quicker the PLL will be able to lock.

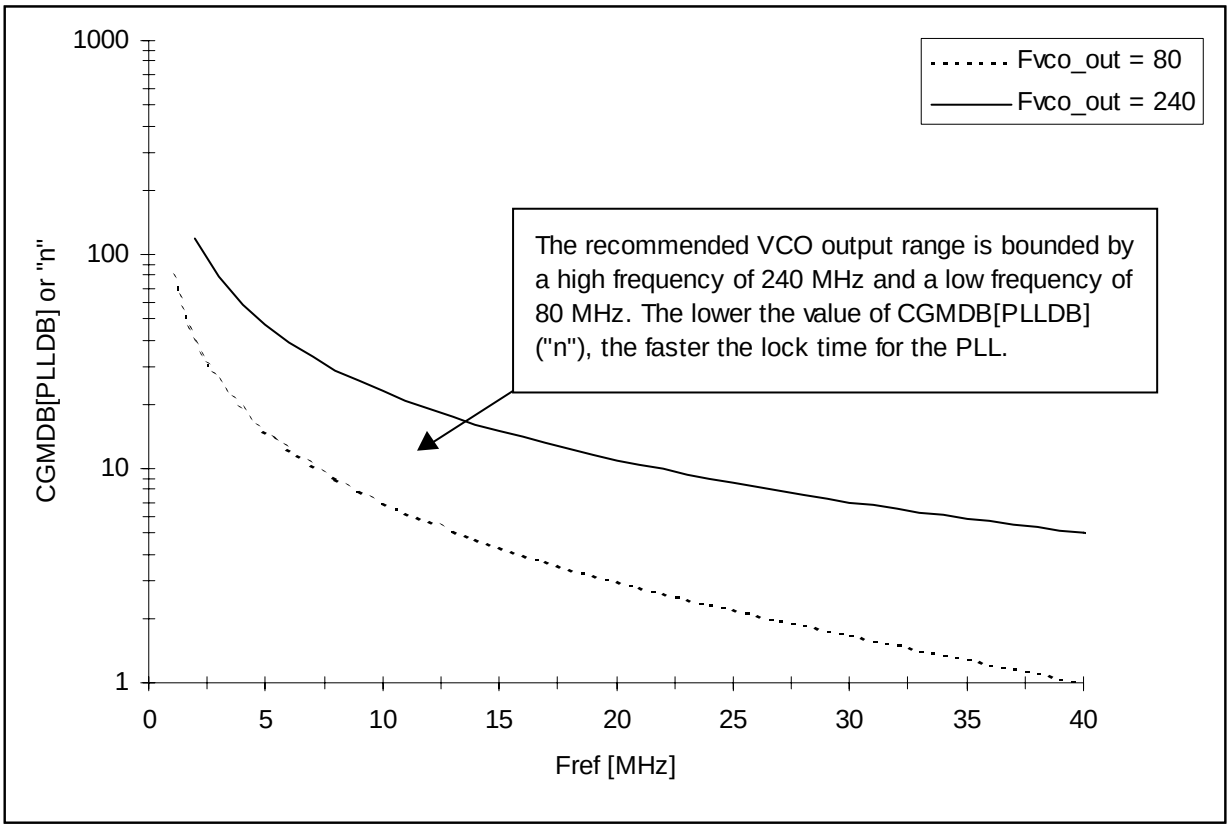


Figure 6-7. PLL Output Frequency vs. Input Frequency

The lock time of the PLL is, in many applications, the most critical PLL design parameter. Proper use of the PLL ensures the highest stability and lowest lock time.

6.3.6.1 PLL Lock Time Determination

Typical control systems refer to the lock time as the reaction time, within specified tolerances, of the system to a step input. In a PLL the step input occurs when the PLL is turned on or when it suffers a noise hit. The tolerance is usually specified as a percent of the step input or when the output settles to the desired value plus or minus a percent of the frequency change. Therefore, the reaction time is constant in this definition regardless of the size of the step input.

When the PLL is coming from a powered down state (PDN is high) to a powered up condition (PDN is low) the maximum lock time is 10msec.

Other systems refer to lock time as the time the system takes to reduce the error between the actual output and the desired output to within specified tolerances. Therefore, the lock time varies according to the original error in the output. Minor errors may be shorter or longer in many cases.

6.3.6.2 PLL Parametric Influences on Reaction Time

Lock time is designed to be as short as possible while still providing the highest possible stability. Many factors directly and indirectly affect the lock time.

The most critical parameter affecting the reaction time of the PLL, is the Reference Frequency (Fref). This frequency is the input to the phase detector and controls how often the PLL makes corrections. For stability, the corrections must be small compared to the desired frequency, so several corrections are required to reduce the frequency error. Therefore, the slower the Reference Frequency (Fref), the longer it takes to make these corrections.

Temperature and processing also can affect acquisition time because the electrical characteristics of the PLL change. The part operates as specified as long as these influences stay within the specified limits.

6.4 CGM Functional Detail

The CGM controls the PLL, detects PLL lock, and is used to generate the master clock to the SIM. The CPU clock is one half the frequency of the master clock while the IPBus clock is one fourth the frequency. The SIM handles these clock divisions. The master clock source can be either the oscillator output or the analog PLL output. The oscillator output (Fref) will typically be 4MHz, but a faster active clock can be driven into the XTAL pin at speeds of up to 240MHz. The PLL output can be up to 240MHz.

In order to use the PLL, the proper divide by factor and post scaler values should be programmed into the CGMDB register. Next, the PLL is turned on by setting the Power-Down (PDN) bit in the CGMCR to zero. The user should then wait for the PLL to achieve lock before changing the SEL bit to select the PLL output as the master clock.

6.4.1 PLL Frequency Lock Detector

This CGM function monitors the VCO output clock and sets the LCK1 and LCK0 bits in the CGM Control register based on the frequency accuracy. The lock detector is enabled with the LCKON bit of the CGMCR as well as the PDN bit. Once enabled, the detector starts two counters whose outputs are periodically compared. The input clocks to these counters are the VCO output clock divided by the divide-by factor, feedback, and the crystal reference clock, Fref. The period of the pulses being compared cover one whole period of each clock because the feedback clock doesn’t guarantee a 50 percent duty cycle.

Counts are compared after 16, 32, and 64 cycles. If the counts match after 32 cycles, the LCK0 bit is set to 1. If the counts match after 64 cycles, the LCK1 bit is also set. The LCK bits stay set until the counts fail to match or if a new value is written to the PLLDB field or on reset caused by LCKON, PDN, or chip level reset. When the circuit sets LCK1, the two counters are reset and

start the count again. The lock detector is designed so if LCK1 is reset to 0 because the counts did not match when checked after 64 cycles, the LCK0 bit can remain high if the counts matched after 32 cycles. This provides the processor the accuracy of the two clocks with respect to each other.

6.5 Module Memory Map

There are three registers on the CGM peripheral described in **Table 6-1.**

Table 6-1. CGM Memory Map $1FFF10

Address Offset	Register Acronym	Register Name	Access Type	Chapter Location
Base + $0	CGMCR	Control Register	Read/Write	Section 6.6.1
Base + $1	CGMDB	Divide-By Register	Read/Write	Section 6.6.2
Base + $2	CGMTOD	Time-of-Day Register	Read/Write	Section 6.6.3

Add. Offset	Register Name		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
$0	CGMCR	R	0	0	LCK1	LCK0	SEL	0	0	0	0	LCK1_IE		LCK0_IE		LCK ON	TOD_ SEL	PDN
		W																
$1	CGMDB	R	POST			0	0	0	0	0	0	PLLDB						
		W																
$2	CGMTOD	R	0	0	0	0	TOD											
		W																

R 0 Read as 0
W Reserved

Figure 6-8. OCCS Register Map Summary

6.6 Register Descriptions (CGM_BASE = $1FFF10)

6.6.1 Clock Generation Module (CGM) Control Register

Base + $0	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	0	0	LCK1	LCK0	SEL	0	0	0	0	LCK1_IE		LCK0_IE		LCKON	TOD_SEL	PDN
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Figure 6-9. CGM Control Register (CGMCR)

See Programmer's Sheet on Appendix page B-14

6.6.1.1 Reserved—Bits 15–14

This bit field is reserved or not implemented. It is read as 0, but cannot be modified by writing.

6.6.1.2 Lock 1 Status (LCK1)—Bit 13

This bit shows the status of the lock detector state for the LCK1 circuit. Changes in the state of this bit can be used to cause interrupts in conjunction with the LCK1 Interrupt Enable bits. The LCK1 interrupt is cleared by writing 1 to this bit.

- 0 = PLL not locked
- 1 = PLL locked

6.6.1.3 Lock 0 Status (LCK0)—Bit 12

This bit shows the status of the lock detector state for the LCK0 circuit. Changes in the state of this bit can be used to cause interrupts in conjunction with the LCK0 Interrupt Enable bits. The LCK0 interrupt is cleared by writing 1 to this bit.

- 0 = PLL not locked
- 1 = PLL locked

6.6.1.4 Clock Source Select (SEL)—Bit 11

This bit is used to control the source of the master clock to the SIM.

- 0 = Oscillator output selected by default
- 1 = PLL output selected

6.6.1.5 Reserved—Bits 10–7

This bit field is reserved or not implemented. It is read as 0, but cannot be modified by writing.

6.6.1.6 Lock 1 Interrupt Enable (LCK1_IE)—Bits 6–5

An optional interrupt can be generated if the PLL lock status bit (LCK1) changes.

- 00 = Disable interrupt by default
- 01 = Enable interrupt on rising edge of LCK1
- 10 = Enable interrupt on falling edge of LCK1
- 11 = Enable interrupt on any edge of LCK1

6.6.1.7 Lock 0 Interrupt Enable (LCK0_IE)—Bits 4–3

An optional interrupt can be generated if the PLL Lock (LCK0) status bit changes.

- 00 = Disable interrupt by default

- 01 = Enable interrupt on rising edge of LCK0
- 10 = Enable interrupt on falling edge of LCK0
- 11 = Enable interrupt on any edge of LCK0

6.6.1.8 Lock Detector On (LCKON)—Bit 2

- 0 = Lock detector disabled by default
- 1= Lock detector enabled

6.6.1.9 Time-of-Day Clock Source Select (TOD_SEL)—Bit 1

This bit is used to select between two possible TIME_CLK sources. The oscillator can generate the TIME_CLK only when the input clock on either the EXTAL or XTAL pins in 4MHz or less. When driving high speed clocks into XTAL, the CGM must generate the TIME_CLK using the CGMTOD register. This bit is only reset by Power-On Reset (POR) conditions.

- 0 = TIME_CLK is generated by oscillator as default
- 1 = TIME_CLK is generated by CGM

6.6.1.10 PLL Power-Down (PDN)—Bit 0

The PLL can be turned off by setting the PDN bit to one. There is a four IPBus clock delay from changing the bit to signaling the PLL. When the PLL is powered down, the clock select logic automatically switches to the oscillator output in order to prevent loss of clock to the core.

- 0 = PLL turned on
- 1 = PLL powered down by default

6.6.2 Clock Generation Module (CGM) Divide-By Register

Base + $1	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	POST			0	0	0	0	0	0	PLLDB						
Write																
Reset	0	0	0	0	0	0	0	0	0	0	1	1	1	0	1	1

Figure 6-10. CGM Divide-By Register (CGMDB)

See Programmer's Sheet on Appendix page B-16

6.6.2.1 PLL Post Scaler (POST)—Bits 15–13

The output of the PLL is postscaler by 1 to 128 based on this field. When changing this field, it is recommended the SEL bit is set to choose the oscillator output, changing this field, then the SEL bit is returned to selecting the PLL's postscaler output.

- 000 = PLL output is divided by one by default
- 001 = PLL output is divided by two

- 010 = PLL output is divided by four
- 011 = PLL output is divided by eight
- 100 = PLL output is divided by 16
- 101 = PLL output is divided by 32
- 110 = PLL output is divided by 64
- 111 = PLL output is divided by 128

6.6.2.2 Reserved—Bits 12–7

This bit field is reserved or not implemented. It is read as 0, but cannot be modified by writing.

6.6.2.3 PLL Divide-By (PLLDB)—Bits 6–0

The VCO output frequency is controlled by the PLL divide-by value. Each time a new value is written into the PLLDB field, the Lock Detector circuit is reset. Before changing the divide-by value, it is recommended the SEL bit be set to choose the oscillator output. The VCO output frequency is determined by using the following formula:

$$Fvco\_out = Fref \times (PLLDB + 1)$$

6.6.3 Clock Generation Module (CGM) Time-of-Day Register

Base + $2	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	0	0	0	0	TOD											
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 6-11. CGM Time-of-Day Register (CGMTOD)

See Programmer's Sheet on Appendix page B-17

6.6.3.1 Reserved—Bits 15–12

This bit field is reserved or not implemented. It is read as 0, but cannot be modified by writing.

6.6.3.2 TOD Scale Factor (TOD)—Bits 11–0

The output of the oscillator is divided by (TOD + 1) and then divided by 2, generating the TIME_CLK used by the COP module when TOD_SEL is high. The value of TOD should be chosen to result in a TOD clock frequency in the range of 15.12KHz to 31.25KHz. This register is only reset during Power-On Reset (POR).

6.7 OCCS Resets

The CGM registers are reset by a chip level reset. This forces all registers to their reset state and selects the oscillator output as the master clock source for the SIM.

6.8 OCCS Interrupts

The CGM generates a single interrupt request to the INTC. This interrupt is generated by the lock detector circuitry LCK0 and LCK1 outputs and is enabled by the LCK0 Interrupt Enable and LCK1 Interrupt Enable bits in the CGMCR. This interrupt can be used to detect when the PLL goes into lock or when it falls out of lock. The interrupt is cleared by writing a 1 to the LCK0 and/or LCK1 bits of the CGMCR.

Chapter 7
Power-On Reset (POR) and Computer Operating Properly (COP)

7.1 Introduction

The Power-on Reset (POR) function monitors the core power supply, the I/O, and analog power supply. If either of those power supplies are below their thresholds, the POR output for each respective supply is held high. Once the power supply goes above the thresholds, the POR outputs are held low.

Computer Operating Properly (COP) is also discussed in this chapter as it relates to resets.

7.2 Features

- The circuit monitors both the core power supply and peripheral power supply
- Holds a wide chip reset once either of these supply voltages are below the thresholds
- Generates the address of the Reset Vector provided to the core after exit Reset
- The address of Reset Vector (same as the COP Reset) is located at $1F0000

The COP module design features include:

- Programmable time-out period = (COP_PRESCALER × (CT + 1)) oscillator clock cycles, where CT can be from $0000 to $FFFF
- Programmable Wait and Stop mode operations
- COP timer is disabled while host CPU is in Debug mode

7.3 Block Diagram

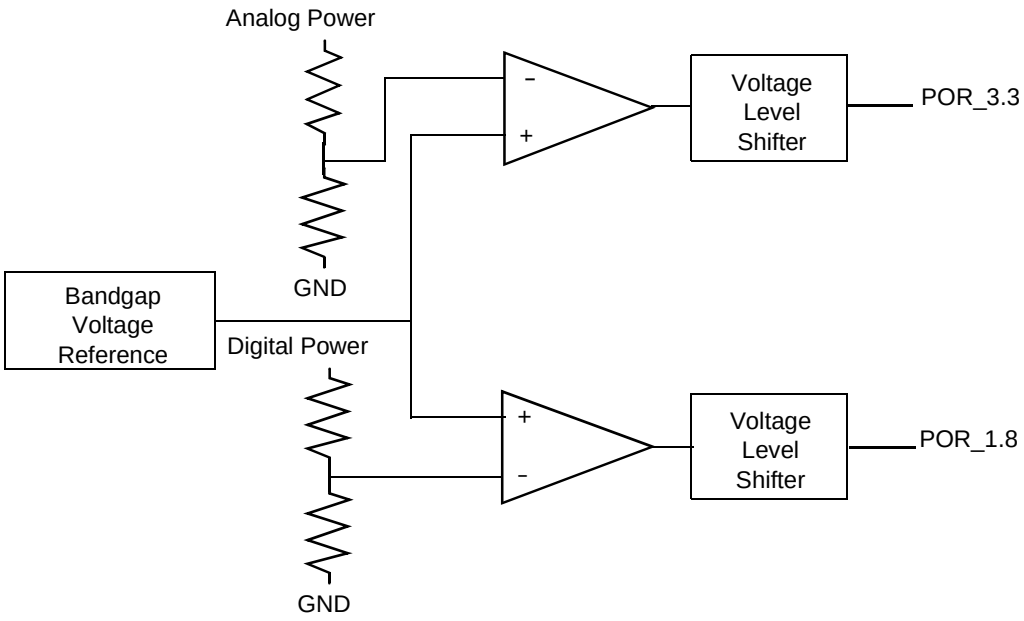


Figure 7-1. POR Module Block Diagram

7.4 Method of Operation

Starting with the chip unpowered, the analog and digital power supplies turned on, the bandgap voltage reference and the comparators will begin to function. The bandgap voltage reference will apply a temperature and supply stable voltage reference to the positive inputs of the comparators. Negative inputs of the comparators are connected to voltage points that move proportionately with respect to the analog and digital power supplies.

Initially, the bandgap voltage reference point is greater than the power supply reference signals and the output of the comparators is high. As each power supply goes above it's trip point, the voltage on the respective comparators negative input will become higher in value than the bandgap voltage reference voltage on the positive input to the comparator and the output of the comparator will go low. If either power supply drops below the trip point the respective POR output will again go high.

For the analog power supply, the POR trip point is:

Absolute Minimum	Nominal	Absolute Maximum
2.8V	2.85V	2.9V

For the digital power supply the POR trip point is:

Absolute Minimum	Nominal	Absolute Maximum
1.62V	1.66V	1.7V

This means, as long as the analog power supply is below 2.8V, the POR_3p3 will be high. When the analog power supply exceeds 2.9V the POR_3p3 output will be low. Respectively, for the digital power supply, when the digital power supply is below 1.62V the POR_1p8 output will be high. When the digital power supply is above 1.7V the POR_1p8 output will be low.

7.5 Computer Operating Properly (COP) Module

The Computer Operating Properly (COP) module is used to help software recover from runaway code. The COP is a free-running down counter, once enabled is designed to generate a Reset upon reaching zero. Software must periodically service the COP in order to clear the counter and prevent a reset.

7.5.1 COP Functional Description

When the COP is enabled, each positive edge of OSCCLK will cause the counter to decrement by one. If the count reaches a value of $0000, then the COP_$\overline{\text{RST}}$ signal is asserted and the chip is reset. In order for the CPU to show it is operating properly, it must perform a service routine prior to the count reaching $0000. The service routine consists of writing $5555 followed by $AAAA to COPCTR.

7.5.2 Time-Out Specifications

The COP uses a 16-bit counter, being clocked by the crystal oscillator clock prescaled by 128. **Table 7-1** presents the range of time-out values supported as a function of oscillator frequency.

Table 7-1. COP Time-out Ranges as a Function of Oscillator Frequency

CT	2 MHz	4MHz
$0000	64 µsec	32 µsec
$FFFF	4.2 sec	2.1 sec

For a crystal operating at 4MHz the clock to the COP counter will be 31.25KHz. The value of the COPTO register can be programmed from 1 to 65535 giving a time-out period range from 32µsec minimum to 2.1sec maximum.

7.5.3 COP After Reset

COPCTL is cleared out of reset. Thus the counter is disabled by default. In addition, COPTO is set to it's maximum value of $FFFF during reset so the counter is loaded with a maximum time-out period when reset is released.

7.5.4 Wait Mode Operation

If both CEN and CWEN are set to 1 and the Wait mode is entered, the COP counter will continue to count down. If either CEN or CWEN is set to 0 when Wait mode is entered, the counter will be disabled and will reload using the value in the COPTO register.

7.5.5 Stop Mode Operation

If both CEN and CSEN set to 1 and the Stop mode is entered, the COP counter will continue to count down. If either CEN or CSEN is set to 0 when Stop mode is entered, the counter will be disabled and will reload using the value in the COPTO register.

7.5.6 Debug Mode Operation

The COP counter does not count when the chip is in the Debug mode. Additionally, the CEN bit in the COPCTL always reads as 0 when the chip is in the Debug mode. The actual value of CEN is unaffected by debug however, and it resumes it's previously set value upon exiting Debug.

7.6 Operating Modes

The COP module design contains two major modes of operation:

- Functional Mode

  The COP by default is in this mode and will remain in this mode for as long as the SCANTESTMODE input remains low.

- Debug Mode

  The COP timer is stopped while the processor is in Debug mode. If the COP is enabled, the timer will resume, counting upon exiting Debug mode. The CEN bit in COPCTL register always reads as 0 when in the Debug mode, even when it has a value of 1.

7.7 Block Diagram

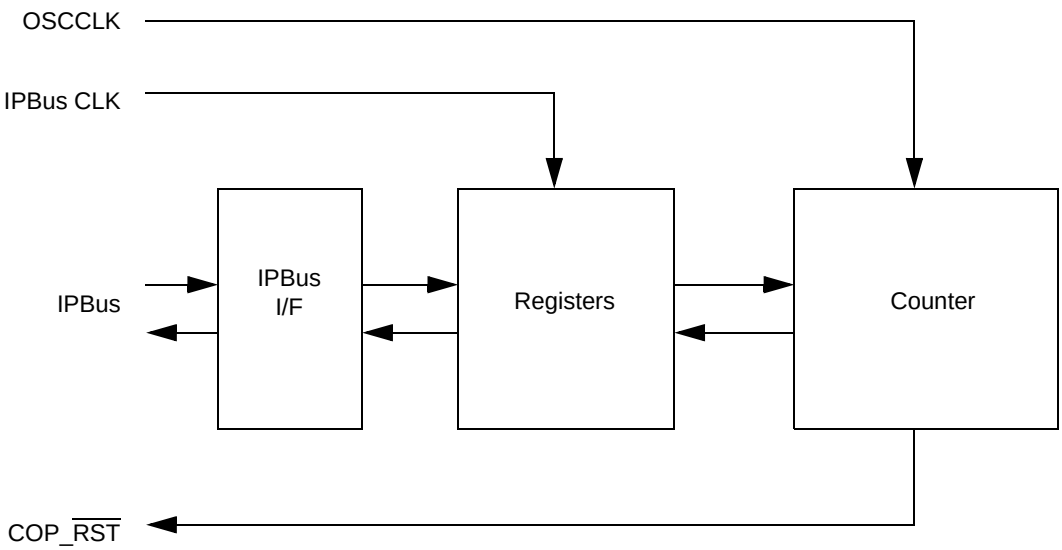


Figure 7-2. COP Module Block Diagram and Interface Signals

7.8 Module Memory Map

There are three registers on the COP peripheral described in **Table 7-2.**

Table 7-2. COP Module Memory Map (COP_BASE = $1FFFD0)

Address Offset	Register Acronym	Register Name	Access Type	Chapter Location
Base + $0	COPCTL	Control Register	Read/Write	**Section 7.9.1**
Base + $1	COPTO	Time-Out Register	Read/Write	**Section 7.9.2**
Base + $2	COPCTR	Counter Register	Read/Write	**Section 7.9.3**

Add. Offset	Register Name		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
$0	COPCTL	R	0	0	0	0	0	0	0	0	0	0	0	BYPS	CSEN	CWEN	CEN	CWP
		W																
$1	COPTO	R	TIMEOUT															
		W																
$2	COPCTR	R	COUNT															
		W	SERVICE															

R 0 Read as 0
W Reserved

Figure 7-3. COP Register Map Summary

7.9 Register Descriptions (COP_BASE = $1FFFD0)

7.9.1 COP Control Register (COPCTL)

Base + $0	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	0	0	0	0	0	0	0	0	0	0	0	BYPS	CSEN	CWEN	CEN	CWP
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 7-4. COP Control Register (COPCTL)

See Programmer's Sheet on Appendix page B-18

7.9.1.1 Reserved—Bits 15–5

This bit field is reserved or not implemented. Each bit in the field is read as 0 and cannot be modified by writing.

7.9.1.2 Bypass (BYPS)—Bit 4

This bit is intended for factory use only. Setting this bit allows testing time of the COP to be accelerated by routing the IPBus clock to the counter instead of the OSCCLK. This bit should not be set during normal operation of the chip. If this bit is used, however, it should only be changed while the CEN bit is set to 0.

- 0 = Counter uses OSCCLK (default)
- 1 = Counter uses IPBus clock

7.9.1.3 COP Stop Mode Enable (CSEN)—Bit 3

This bit controls the operation of the COP counter in the Stop mode. This bit can only be changed when the CWP bit is set to 0.

- 0 = COP counter will stop in the Stop mode (default)
- 1 = COP counter will run in the Stop mode if CEN is set to 1

7.9.1.4 COP Wait Mode Enable (CWEN)—Bit 2

This bit controls the operation of the COP counter in the Wait mode. This bit can only be changed when the CWP bit is set to 0.

- 0 = COP counter will stop in the Wait mode (default)
- 1 = COP counter will run in the Wait mode if CEN is set to 1

7.9.1.5 COP Enable (CEN)—Bit 1

This bit controls the operation of the COP counter. This bit can only be changed when the CWP bit is set to 0. This bit always reads as 0 when the chip is in the Debug mode.

- 0 = COP counter is disabled (default)
- 1 = COP counter is enabled

7.9.1.6 COP Write Protect (CWP)—Bit 0

This bit controls the write protection feature of the COP Control (COPCTL) register and the COP Time-Out (COPTO) register. Once set, this bit can only be cleared by resetting the module.

- 0 = COPCTL and COPTO are readable and writable (default)
- 1 = COPCTL and COPTO are read only

7.9.2 COP Time-Out Register (COPTO)

Base + $1	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	TIMEOUT															
Write																
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Figure 7-5. COP Time-Out Register (COPTO)

See Programmer's Sheet on Appendix page B-19

7.9.2.1 COP Time-Out Period (TIMEOUT)—Bits 15–0

The value in this register determines the time-out period of the COP counter. TIMEOUT should be written before the COP is enabled. Once the COP is enabled, the recommended procedure for changing TIMEOUT is to disable the COP, write to COPTO, then re-enable the COP, ensuring the new TIMEOUT is loaded into the counter. Alternatively, the CPU can write to COPTO, then write the proper patterns to COPCTR, causing the counter to reload with the new TIMEOUT value. The COP counter is not reset by a write to COPTO. Changing TIMEOUT while the COP is enabled will result in a time-out period differing from the expected value. These bits can only be changed when the CWP bit is set to 0.

7.9.3 COP Counter Register (COPCTR)

Base + $2	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	COUNT															
Write	SERVICE															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Figure 7-6. COP Counter Register (COPCTR)

See Programmer's Sheet on Appendix page B-20

7.9.3.1 COP Count (COUNT)—Bits 15–0

This is the current value of the COP counter as it counts down from the time-out value to zero. A reset is issued when this count reaches zero.

7.9.3.2 COP Service (SERVICE)—Bits 15–0

When enabled, the COP requires a service sequence be performed periodically in order to clear the COP counter and prevent a reset from being issued. This routine consists of writing $5555 to the COPCTR followed by writing $AAAA before the time-out period expires. The writes to COPCTR must be performed in the correct order, but any number of other instructions, and writes to other registers, may be executed between the two writes.

7.10 Clocks

The COP timer base is the oscillator clock divided by a fixed prescalar value. The prescalar divisor for this chip is 128. All register timing is with regard to the IPBus clock.

7.11 Resets

Any system reset forces all registers to their reset state, clearing the $\overline{\text{RST}}$ signal when it is asserted. The counter will be loaded with its maximum value of $FFFF, but it will not start when Reset is released because the CEN bit is disabled by default.

7.12 Interrupts

The COP module does not generate any interrupts. It does generate the $\overline{\text{RST}}$ signal when the counter reaches a value of $0000, causing a chip wide reset.

Chapter 8
Interrupt Controller (ITCN)

8.1 Introduction

The Interrupt Controller (ITCN) is responsible for arbitrating all interrupt requests according to the priority level of the each request. This includes all external interrupt sources, such as $\overline{\text{IRQA}}$, $\overline{\text{IRQB}}$, and so on, peripheral generated interrupt requests and core generated interrupt requests. After arbitration, the interrupt controller will compare the priority of the current interrupt request with the current priority level of the core and if the request has higher priority to generate a single enabled interrupt request signal to the core.

There are five levels of interrupt priority provided by the 56800E core, illustrated in **Table 8-1:**

1. LP = the lowest level is generated by the SWILP instruction
2. Level 0 = maskable with the lowest priority of the three maskable interrupts
3. Level 1 = maskable
4. Level 2 = maskable
5. Level 3 = the highest priority is a non-maskable interrupt

Device interrupt priority levels are programmable via the Interrupt Priority Register (IPR).

The interrupt controller is also responsible for generating the vector address of the current interrupt request. This is based on the Vector Address Base (VAB) register and the event initiating the request. The interrupt controller predefines the Vector Table offsets for all possible interrupt sources and will generate the Vector Address of the request by adding the programmable VAB register to the Vector Table offset.

Table 8-1. Interrupt Priority Level

IPL	Description	Priority	Interrupt Sources
LP	Maskable	Lowest	SWILP instruction
0	Maskable	—	On-chip peripherals, IRQA and IRQB, SWI #0 instruction
1	Maskable	—	On-chip peripherals, IRQA and IRQB, SWI #1 instruction, EOnCE interrupts
2	Maskable	—	On-chip peripherals, IRQA and IRQB, SWI #2 instruction, EOnCE interrupts
3	Non-maskable	Highest	Illegal instruction, HWS overflow, SWI #3 instruction, EOnCE interrupts, Misaligned data access

External interrupt sources such as $\overline{\text{IRQA}}$ and $\overline{\text{IRQB}}$ are programmable to either level sensitive or edge triggered. Level sensitive interrupts remain active as long as the input remains low and are cleared when the input level goes high. The edge sensitive interrupts are latched as pending on the high-to-low transition of the interrupt input and are cleared when the interrupt is serviced.

The Vector Table is structured as two words per vector. This implies the Interrupt Vector offset, added to the Vector Base Address, will be the vector number multiplied-by two. If the first instruction of an interrupt vector is a JSR or BSR, the core assumes a standard long interrupt. This is the normal case.

The core then saves the status register and program counter and vectors to the address pointed to by the JSR. The interrupt is cleared by executing the return from interrupt instruction (RTI or RTID) at the end of the interrupt service routine.

The interrupt controller can also support up to two fast interrupts. There are four programmable registers in the controller, two for each fast interrupt, allowing set-up of a vector number to be configured as a fast interrupt, and a 21-bit absolute vector address pointing to the interrupt service routine.

When the Fast Interrupt Vector Number register is programmed, the interrupt controller will intercept the normal vector table processing and insert the absolute address into the core via the VAB bus. As long as the first instruction of the interrupt service routine is not a JSR or a BSR, the core will interpret the interrupt as a fast interrupt and begin inserting the code into the pipeline until a Fast Return from Interrupt (FRTID) is executed. The interrupt priority must be set to level two for the fast interrupt to operate properly. Further, there can not be a JSR or BSR as the first instruction of the fast interrupt service routine. The return from interrupt must use the Fast Return from Interrupt (FRTID) instruction to clear the interrupt.

Reset is considered to be the highest priority interrupt and will take precedence over all other interrupts. If the Reset pin is pulled low the interrupt controller will generate a reset vector address for the core and assert the re-signal in the core.

8.2 Features

The ITCN module design includes these capabilities:

- Programmable priority levels for each IRQ
- Two programmable Fast Interrupts
- Notification to SIM module to restart clocks out of Wait and Stop modes

8.3 Signal Description

The ITCN module interfaces with the IPBus, the 56800E core, and the IRQ sources. There are no chip outputs driven directly by this module, but the $\overline{\text{IRQA}}$ and $\overline{\text{IRQB}}$ chip inputs do come to the ITCN where they are re-synchronized to the system clock before use.

8.4 Module Memory Map

There are 22 registers on the ITCN peripheral described in **Table 8-2.**

Table 8-2. ITCN Module Memory Map (ITCN_BASE = $1FFF20)

Address Offset	Register Acronym	Register Name	Access Type	Chapter Location
Base + $0	IPR0	Interrupt Priority Register 0	Read/Write	Section 8.7.1
Base + $1	IPR1	Interrupt Priority Register 1	Read/Write	Section 8.7.2
Base + $2	IPR2	Interrupt Priority Register 2	Read/Write	Section 8.7.3
Base + $3	IPR3	Interrupt Priority Register 3	Read/Write	Section 8.7.4
Base + $4	IPR4	Interrupt Priority Register 4	Read/Write	Section 8.7.5
Base + $5	IPR5	Interrupt Priority Register 5	Read/Write	Section 8.7.6
Base + $6	IPR6	Interrupt Priority Register 6	Read/Write	Section 8.7.7
Base + $7	IPR7	Interrupt Priority Register 7	Read/Write	Section 8.7.8
Base + $8	IPR8	Iinterrupt Priority Register 8	Read/Write	Section 8.7.9
Base + $9	VBA	Vector Base Address Register	Read/Write	Section 8.7.10
Base + $A	FIM0	Fast Interrupt Match Register 0	Read/Write	Section 8.7.11
Base + $B	FIVAL0	Fast Interrupt Vector Address Low 0	Read/Write	Section 8.7.12
Base + $C	FIVAH0	Fast Interrupt Vector Address High 0	Read/Write	
Base + $D	FIM1	Fast Interrupt Match Register 1	Read/Write	Section 8.7.11
Base + $E	FIVAL1	Fast Interrupt Vector Address Low 1	Read/Write	Section 8.7.13
Base + $F	FIVAH1	Fast Interrupt Vector Address High 1	Read/Write	Section 8.7.14
Base + $10	IRQP0	IRQ Pending Register 0	*Read Only*	Section 8.7.15
Base + $11	IRQP1	IRQ Pending Register 1	*Read Only*	
Base + $12	IRQP2	IRQ Pending Register 2	*Read Only*	
Base + $13	IRQP3	IRQ Pending Register 3	*Read Only*	
Base + $14	IRQP4	IRQ Pending Register 4	*Read Only*	
Base + $1A	ICTL	Interrupt Control Register	Read/Write	Section 8.7.16

Add. Offset	Register Name		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
$0	IPR0	R	0	0	BKPT_U0 IPL		STPCNT IPL		0	0	0	0	0	0	0	0	0	0
		W																
$1	IPR1	R	0	0	0	0	0	0	0	0	0	0	RX_REG IPL		TX_REG IPL		TRBUF IPL	
		W																
$2	IPR2	R	DMA2 IPL		DMA1 IPL		DMA0 IPL		0	0	LOCK IPL		0	0	IRQB IPL		IRQA IPL	
		W																
$3	IPR3	R	SS0_TD IPL		SS0_TDES IPL		SS0_RLS IPL		SS0_RD IPL		SS0_RDES IPL		DMA5 IPL		DMA4 IPL		DMA3 IPL	
		W																
$4	IPR4	R	SPI_RCV IPL		ESSI1_TLS IPL		ESSI1_TD IPL		ESSI1_TDES IPL		ESSI1_RLS IPL		ESSI1_RD IPL		ESSI1_IPL		ESSI0_TLS IPL	
		W																
$5	IPR5	R	HOST_XMIT IPL		HOST_RCV IPL		SCI0_RCV IPL		SCI0_RERR IPL		SCI0_RIDL IPL		SCI0_TIDL IPL		SCI0_XMIT IPL		SPI_XMIT IPL	
		W																
$6	IPR6	R	TOVF1 IPL		TCMP1 IPL		TINP0 IPL		TOVF0 IPL		TCMP0 IPL		TOD_IPL		TOD_ALRM IPL		HOST_CMD IPL	
		W																
$7	IPR7	R	0	0	TINP3 IPL		TOVS3 IPL		TCMP3 IPL		TINP2 IPL		TOVF2 IPL		TCMP2 IPL		TINP1 IPL	
		W																
$8	IPR8	R	0	0	0	0	0	0	SCI1_RCV IPL		SCI1_RERR IPL		SCI1_RIDL IPL		SCI1_TIDL IPL		SCI1_XMIT IPL	
		W																
$9	VBA	R	0	0	0	VECTOR_BASE_ADDRESS												
		W																
$A	FIM0	R	0	0	0	0	0	0	0	0	0	FAST INTERRUPT 0						
		W																
$D	FIM1	R	0	0	0	0	0	0	0	0	0	FAST INTERRUPT 1						
		W																
$B	FIVAL0	R	FAST INTERRUPT 0 VECTOR ADDRESS LOW															
		W																
$C	FIVAH0	R	0	0	0	0	0	0	0	0	0	0	0	FAST INTERRUPT 0 VECTOR ADDRESS HIGH				
		W																
$E	FIVAL1	R	FAST INTERRUPT 1 VECTOR ADDRESS LOW															
		W																
$F	FIVAH1	R	0	0	0	0	0	0	0	0	0	0	0	FAST INTERRUPT1 VECTOR ADDRESS HIGH				
		W																
$10	IRQP0	R	PENDING [16:2]															1
		W																
$11	IRQP1	R	PENDING [32:17]															
		W																
$12	IRQP2	R	PENDING [48:33]															
		W																
$13	IRQP3	R	PENDING [64:49]															
		W																
$14	IRQP4	R	1	1	1	1	1	1	1	1	1	1	1	PENDING [69:65]				
		W																
$1A	ICTL	R	INT	IPIC			VAB						INT_DIS	0	IRQB STATE	IRQA STATE	IRQB EDG	IRQA EDG
		W																

R	0	Read as 0
W		Reserved

Figure 8-1. ITCN Register Map Summary

8.5 Block Diagram

The ITCN module block diagram is illustrated **Figure 8-2.**

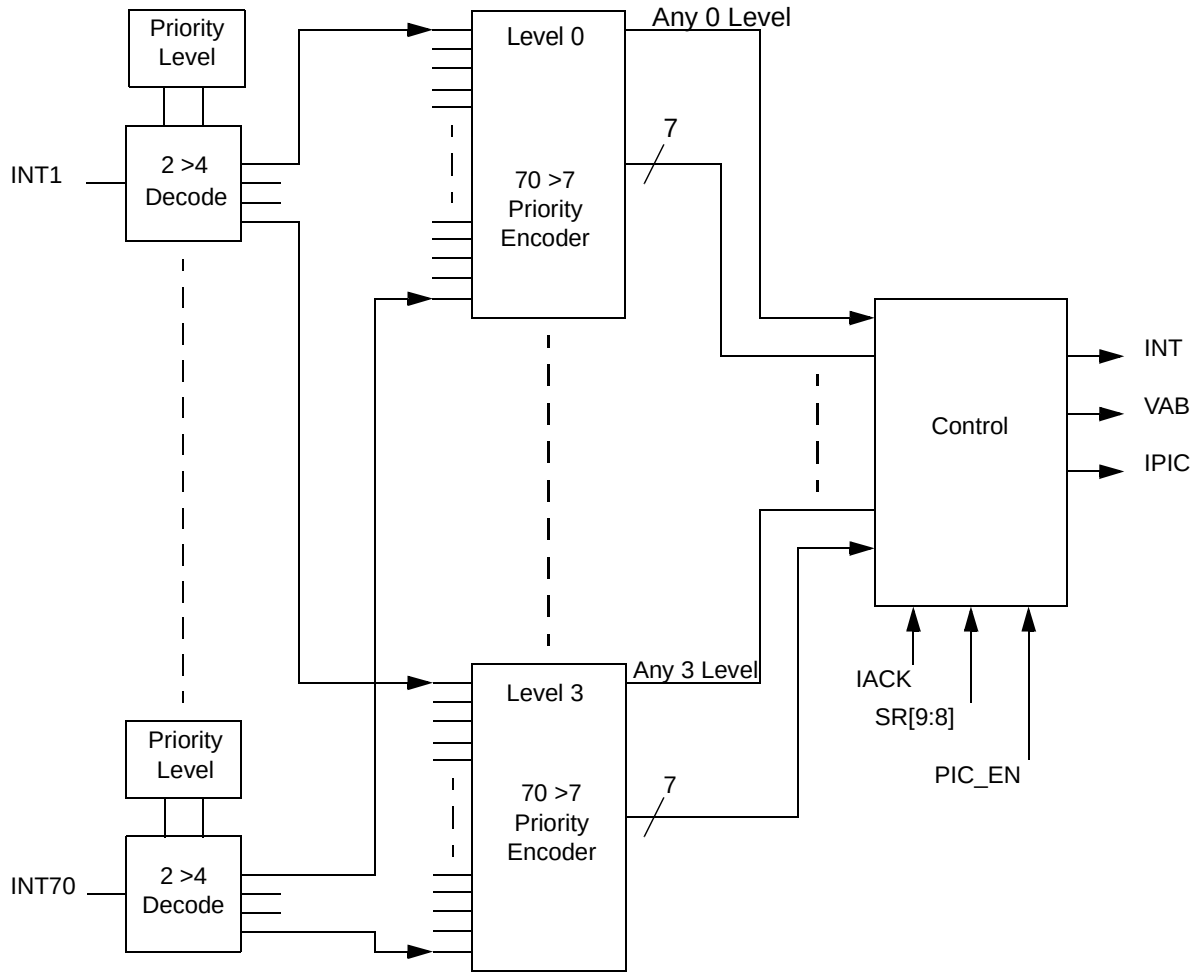


Figure 8-2. Interrupt Controller Block Diagram

8.6 Functional Description

The Interrupt Controller is a slave on the IPBus. It contains registers each allowing the 70 *interrupt sources* to be set to one of four priority levels (excluding certain interrupts of fixed priority). Next, all of the interrupt requests of a given level are priority encoded to determine the lowest numerical value of the active interrupt requests for that level. Within a given priority level, zero is the highest priority and number 69 is the lowest.

8.7 Register Descriptions (ITCN_BASE = $1FFF20)

8.7.1 Interrupt Priority Register 0 (IPR0)

Base + $0	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	0	0	BKPT_U0 IPL		STPCNT IPL		0	0	0	0	0	0	0	0	0	0
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 8-3. Interrupt Priority Register 0 (IPR0)

See Programmer's Sheet on Appendix page B - 21

8.7.1.1 Reserved—Bits 15–14

These bits are reserved or not implemented. They are read as 0, but they cannot be modified by writing.

8.7.1.2 EOnCE Breakpoint Unit 0 Interrupt Priority Level (BKPT_U0 IPL)—Bits 13–12

This bit field is used to set the interrupt priority levels for certain EOnCE IRQs. These IRQs are limited to priorities one through three and are disabled by default.

- 00 = IRQ disabled (default)
- 01 = IRQ is priority level 1
- 10 = IRQ is priority level 2
- 11 = IRQ is priority level 3

8.7.1.3 EOnCE Step Counter Interrupt Priority Level (STPCNT IPL)—Bits 11–10

This bit field is used to set the interrupt priority levels for certain EOnCE IRQs. These IRQs are limited to priorities one through three and are disabled by default.

- 00 = IRQ disabled (default)
- 01 = IRQ is priority level 1
- 10 = IRQ is priority level 2
- 11 = IRQ is priority level 3

8.7.1.4 Reserved—Bits 9–0

These bits are reserved or not implemented. They are read as 0, but they cannot be modified by writing.

8.7.2 Interrupt Priority Register 1 (IPR1)

Base + $1	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	0	0	0	0	0	0	0	0	0	0	RX_REG IPL		TX_REG IPL		TRBUF IPL	
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 8-4. Interrupt Priority Register 1 (IPR1)

See Programmer's Sheet on Appendix page B - 22

8.7.2.1 Reserved—Bits 15–6

These bits are reserved or not implemented. They are read as 0, but they cannot be modified by writing.

8.7.2.2 EOnCE Receive Register Empty Interrupt Priority Level (RX_REG IPL)—Bits 5–4

This bit field is used to set the interrupt priority levels for certain EOnCE IRQs. These IRQs are limited to priorities one through three and are disabled by default.

- 00 = IRQ disabled (default)
- 01 = IRQ is priority level 1
- 10 = IRQ is priority level 2
- 11 = IRQ is priority level 3

8.7.2.3 EOnCE Transmit Register Full Interrupt Priority Level (TX_REG IPL)—Bits 3–2

This bit field is used to set the interrupt priority levels for certain EOnCE IRQs. These IRQs are limited to priorities one through three and are disabled by default.

- 00 = IRQ disabled (default)
- 01 = IRQ is priority level 1
- 10 = IRQ is priority level 2
- 11 = IRQ is priority level 3

8.7.2.4 EOnCE Trace Buffer Interrupt Priority Level (TRBUF IPL)—Bits 1–0

This bit field is used to set the interrupt priority levels for certain EOnCE IRQs. These IRQs are limited to priorities one through three and are disabled by default.

- 00 = IRQ disabled (default)
- 01 = IRQ is priority level 1
- 10 = IRQ is priority level 2

- 11 = IRQ is priority level 3

8.7.3 Interrupt Priority Register 2 (IPR2)

Base + $2	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	DMA2 IPL		DMA1 IPL		DMA0 IPL		0	0	LOCK IPL		0	0	IRQB IPL		IRQA IPL	
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 8-5. Interrupt Priority Register 2 (IPR2)

See Programmer's Sheet on Appendix page B - 23

8.7.3.1 DMA 2 Done Interrupt Priority Level (DMA2 IPL)—Bits 15–14

This bit field is used to set the interrupt priority levels for certain EOnCE IRQs. These IRQs are limited to priorities one through three and are disabled by default.

- 00 = IRQ disabled (default)
- 01 = IRQ is priority level 0
- 10 = IRQ is priority level 1
- 11 = IRQ is priority level 2

8.7.3.2 DMA 1 Done Interrupt Priority Level (DMA1 IPL)—Bits 13–12

This bit field is used to set the interrupt priority levels for certain EOnCE IRQs. These IRQs are limited to priorities one through three and are disabled by default.

- 00 = IRQ disabled (default)
- 01 = IRQ is priority level 0
- 10 = IRQ is priority level 1
- 11 = IRQ is priority level 2

8.7.3.3 DMA 0 Done Interrupt Priority Level (DMA0 IPL)—Bits 11–10

This bit field is used to set the interrupt priority levels for certain EOnCE IRQs. These IRQs are limited to priorities one through three and are disabled by default.

- 00 = IRQ disabled (default)
- 01 = IRQ is priority level 0
- 10 = IRQ is priority level 1
- 11 = IRQ is priority level 2

8.7.3.4 Reserved—Bits 9–8

These bits are reserved, or are not implemented. They are read as 0, but they cannot be modified by writing.

8.7.3.5 PLL Loss of Lock Interrupt Priority Level (LOCK IPL)—Bits 7–6

This bit field is used to set the interrupt priority levels for certain EOnCE IRQs. These IRQs are limited to priorities one through three and are disabled by default.

- 00 = IRQ disabled (default)
- 01 = IRQ is priority level 0
- 10 = IRQ is priority level 1
- 11 = IRQ is priority level 2

8.7.3.6 Reserved—Bits 5–4

These bits are reserved, or are not implemented. They are read as 0, but cannot be modified by writing.

8.7.3.7 External IRQ B Interrupt Priority Level (IRQB IPL)—Bits 3–2

This bit field is used to set the interrupt priority levels for certain EOnCE IRQs. These IRQs are limited to priorities one through three and are disabled by default.

- 00 = IRQ disabled (default)
- 01 = IRQ is priority level 0
- 10 = IRQ is priority level 1
- 11 = IRQ is priority level 2

8.7.3.8 External IRQ A Interrupt Priority Level (IRQA IPL)—Bits 1–0

This bit field is used to set the interrupt priority levels for certain peripheral IRQs. These IRQs are limited to priorities zero through two and are disabled by default.

- 00 = IRQ disabled (default)
- 01 = IRQ is priority level 0
- 10 = IRQ is priority level 1
- 11 = IRQ is priority level 2

8.7.4 Interrupt Priority Register 3 (IPR3)

Base + $3	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	ESS0_TD IPL		ESS0_TDES IPL		ESS0_RLS IPL		ESS0_RD IPL		ESS0_RDES IPL		DMA5 IPL		DMA4 IPL		DMA3 IPL	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 8-6. Interrupt Priority Register 3 (IPR3)

See Programmer's Sheet on Appendix page B - 25

8.7.4.1 ESSI0 Transmit Data Interrupt Priority Level (ESSI0_TD IPL)—Bits 15–14

This bit field is used to set the interrupt priority levels for certain peripheral IRQs. These IRQs are limited to priorities zero through two and are disabled by default.

- 00 = IRQ disabled (default)
- 01 = IRQ is priority level 0
- 10 = IRQ is priority level 1
- 11 = IRQ is priority level 2

8.7.4.2 ESSI0 Transmit Data with Exception Status Interrupt Priority Level (ESSI0_TDES IPL)—Bits 13–12

This bit field is used to set the interrupt priority levels for certain peripheral IRQs. These IRQs are limited to priorities zero through two and are disabled by default.

- 00 = IRQ disabled (default)
- 01 = IRQ is priority level 0
- 10 = IRQ is priority level 1
- 11 = IRQ is priority level 2

8.7.4.3 ESSI0 Receive Last Slot Interrupt Priority Level (ESSI0_RLS IPL)—Bits 11–10

This bit field is used to set the interrupt priority levels for certain peripheral IRQs. These IRQs are limited to priorities zero through two and are disabled by default.

- 00 = IRQ disabled (default)
- 01 = IRQ is priority level 0
- 10 = IRQ is priority level 1
- 11 = IRQ is priority level 2

8.7.4.4 ESSI0 Receive Data Interrupt Priority Level (ESSI0_RD IPL)—Bits 9–8

This bit field is used to set the interrupt priority levels for certain peripheral IRQs. These IRQs are limited to priorities zero through two and are disabled by default.

- 00 = IRQ disabled (default)
- 01 = IRQ is priority level 0
- 10 = IRQ is priority level 1
- 11 = IRQ is priority level 2

8.7.4.5 ESSI0 Receive Data with Exception Status Interrupt Priority Level (ESSI0_RDES IPL)—Bits 7–6

This bit field is used to set the interrupt priority levels for certain peripheral IRQs. These IRQs are limited to priorities zero through two and are disabled by default.

- 00 = IRQ disabled (default)
- 01 = IRQ is priority level 0
- 10 = IRQ is priority level 1
- 11 = IRQ is priority level 2

8.7.4.6 DMA 5 Done Interrupt Priority Level (DMA5 IPL)—Bits 5–4

This bit field is used to set the interrupt priority levels for certain peripheral IRQs. These IRQs are limited to priorities zero through two and are disabled by default.

- 00 = IRQ disabled (default)
- 01 = IRQ is priority level 0
- 10 = IRQ is priority level 1
- 11 = IRQ is priority level 2

8.7.4.7 DMA 4 Done Interrupt Priority Level (DMA4 IPL)—Bits 3–2

This bit field is used to set the interrupt priority levels for certain peripheral IRQs. These IRQs are limited to priorities zero through two and are disabled by default.

- 00 = IRQ disabled (default)
- 01 = IRQ is priority level 0
- 10 = IRQ is priority level 1
- 11 = IRQ is priority level 2

8.7.4.8 DMA 3 Done Interrupt Priority Level (DMA3 IPL)—Bits 1–0

This bit field is used to set the interrupt priority levels for certain peripheral IRQs. These IRQs are limited to priorities zero through two and are disabled by default.

- 00 = IRQ disabled (default)
- 01 = IRQ is priority level 0
- 10 = IRQ is priority level 1
- 11 = IRQ is priority level 2

8.7.5 Interrupt Priority Register 4 (IPR4)

Base + $4	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	SPI_RCV IPL		ESSI1_TLS IPL		ESSI1_TD IPL		ESSI1_TDES IPL		ESSI1_RLS IPL		ESSI1_RD IPL		ESSI1_RDES IPL		ESSI0_TLS IPL	
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 8-7. Interrupt Priority Register 4 (IPR4)

See Programmer's Sheet on Appendix page B - 27

8.7.5.1 SPI Receiver Full Interrupt Priority Level (SPI_RCV IPL)—Bits 15–14

This bit field is used to set the interrupt priority levels for certain peripheral IRQs. These IRQs are limited to priorities zero through two and are disabled by default.

- 00 = IRQ disabled (default)
- 01 = IRQ is priority level 0
- 10 = IRQ is priority level 1
- 11 = IRQ is priority level 2

8.7.5.2 ESSI1 Transmit Last Slot Interrupt Priority Level (ESSI1_TLS IPL)—Bits 13–12

This bit field is used to set the interrupt priority levels for certain peripheral IRQs. These IRQs are limited to priorities zero through two and are disabled by default.

- 00 = IRQ disabled (default)
- 01 = IRQ is priority level 0
- 10 = IRQ is priority level 1
- 11 = IRQ is priority level 2

8.7.5.3 ESSI1 Transmit Data Interrupt Priority Level (ESSI1_TD IPL)—Bits 11–10

This bit field is used to set the interrupt priority levels for certain peripheral IRQs. These IRQs are limited to priorities zero through two and are disabled by default.

- 00 = IRQ disabled (default)
- 01 = IRQ is priority level 0
- 10 = IRQ is priority level 1
- 11 = IRQ is priority level 2

8.7.5.4 ESSI1 Transmit Data with Exception Status Interrupt Priority Level (ESSI1_TDES IPL)—Bits 9–8

This bit field is used to set the interrupt priority levels for certain peripheral IRQs. These IRQs are limited to priorities zero through two and are disabled by default.

- 00 = IRQ disabled (default)
- 01 = IRQ is priority level 0
- 10 = IRQ is priority level 1
- 11 = IRQ is priority level 2

8.7.5.5 ESSI1 Receive Last Slot Interrupt Priority Level (ESSI1_RLS IPL)—Bits 7–6

This bit field is used to set the interrupt priority levels for certain peripheral IRQs. These IRQs are limited to priorities zero through two and are disabled by default.

- 00 = IRQ disabled (default)
- 01 = IRQ is priority level 0
- 10 = IRQ is priority level 1
- 11 = IRQ is priority level 2

8.7.5.6 ESSI1 Receive Data Interrupt Priority Level (ESSI1_RD IPL)—Bits 5–4

This bit field is used to set the interrupt priority levels for certain peripheral IRQs. These IRQs are limited to priorities zero through two and are disabled by default.

- 00 = IRQ disabled (default)
- 01 = IRQ is priority level 0
- 10 = IRQ is priority level 1
- 11 = IRQ is priority level 2

8.7.5.7 ESSI1 Receive Data with Exception Status Interrupt Priority Level (ESSI1_RDES IPL)—Bits 3–2

This bit field is used to set the interrupt priority levels for certain peripheral IRQs. These IRQs are limited to priorities zero through two and are disabled by default.

- 00 = IRQ disabled (default)
- 01 = IRQ is priority level 0
- 10 = IRQ is priority level 1
- 11 = IRQ is priority level 2

8.7.5.8 ESSI0 Transmit Last Slot Interrupt Priority Level (ESSI0_TLS IPL)—Bits 1–0

This bit field is used to set the interrupt priority levels for certain peripheral IRQs. These IRQs are limited to priorities zero through two and are disabled by default.

- 00 = IRQ disabled (default)
- 01 = IRQ is priority level 0
- 10 = IRQ is priority level 1
- 11 = IRQ is priority level 2

8.7.6 Interrupt Priority Register 5 (IPR5)

Base + $5	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	HOST_XMIT IPL		HOST_RCV IPL		SCI0_RCV IPL		SCI0_RERR IPL		SCI0_RIDL IPL		SCI0_TIDL IPL		SCI0_XMIT IPL		SPI_XMIT IPL	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 8-8. Interrupt Priority Register 5 (IPR5)

See Programmer's Sheet on Appendix page B - 29

8.7.6.1 Host I/F Transmit Interrupt Priority Level (HOST_XMIT IPL)—Bits 15–14

This bit field is used to set the interrupt priority levels for certain peripheral IRQs. These IRQs are limited to priorities zero through two and are disabled by default.

- 00 = IRQ disabled (default)
- 01 = IRQ is priority level 0
- 10 = IRQ is priority level 1
- 11 = IRQ is priority level 2

8.7.6.2 Host I/F Receive Data Interrupt Priority Level (HOST_RCV IPL)—Bits 13–12

This bit field is used to set the interrupt priority levels for certain peripheral IRQs. These IRQs are limited to priorities zero through two and are disabled by default.

- 00 = IRQ disabled (default)
- 01 = IRQ is priority level 0
- 10 = IRQ is priority level 1
- 11 = IRQ is priority level 2

8.7.6.3 SCI 0 Receiver Full Interrupt Priority Level (SCI0_RCV IPL)—Bits 11–10

This bit field is used to set the interrupt priority levels for certain peripheral IRQs. These IRQs are limited to priorities zero through two and are disabled by default.

- 00 = IRQ disabled (default)
- 01 = IRQ is priority level 0
- 10 = IRQ is priority level 1
- 11 = IRQ is priority level 2

8.7.6.4 SCI0 Receiver Error Interrupt Priority Level (SCI0_RERR IPL)—Bits 9–8

This bit field is used to set the interrupt priority levels for certain peripheral IRQs. These IRQs are limited to priorities zero through two and are disabled by default.

- 00 = IRQ disabled (default)
- 01 = IRQ is priority level 0
- 10 = IRQ is priority level 1
- 11 = IRQ is priority level 2

8.7.6.5 SCI0 Receiver Idle Interrupt Priority Level (SCI0_RIDL IPL)—Bits 7–6

This bit field is used to set the interrupt priority levels for certain peripheral IRQs. These IRQs are limited to priorities zero through two and are disabled by default.

- 00 = IRQ disabled (default)
- 01 = IRQ is priority level 0
- 10 = IRQ is priority level 1
- 11 = IRQ is priority level 2

8.7.6.6 SCI0 Transmitter Idle Interrupt Priority Level (SCI0_TIDL IPL)—Bits 5–4

This bit field is used to set the interrupt priority levels for certain peripheral IRQs. These IRQs are limited to priorities zero through two and are disabled by default.

- 00 = IRQ disabled (default)
- 01 = IRQ is priority level 0
- 10 = IRQ is priority level 1
- 11 = IRQ is priority level 2

8.7.6.7 SCI0 Transmitter Empty Interrupt Priority Level (SCI0_XMIT IPL)—Bits 3–2

This bit field is used to set the interrupt priority levels for certain peripheral IRQs. These IRQs are limited to priorities zero through two and are disabled by default.

- 00 = IRQ disabled (default)
- 01 = IRQ is priority level 0
- 10 = IRQ is priority level 1
- 11 = IRQ is priority level 2

8.7.6.8 SPI Transmitter Empty Interrupt Priority Level (SPI_XMIT IPL)—Bits 1–0

This bit field is used to set the interrupt priority levels for certain peripheral IRQs. These IRQs are limited to priorities zero through two and are disabled by default.

- 00 = IRQ disabled (default)
- 01 = IRQ is priority level 0
- 10 = IRQ is priority level 1
- 11 = IRQ is priority level 2

8.7.7 Interrupt Priority Register 6 (IPR6)

Base + $6	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	TOVF1 IPL		TCMP1 IPL		TINP0 IPL		TOVF0 IPL		TCMP0 IPL		TOD IPL		TOD_ALRM IPL		HOST_CMD IPL	
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 8-9. Interrupt Priority Register 6 (IPR6)

See Programmer's Sheet on Appendix page B - 31

8.7.7.1 Timer Channel 1 Overflow Interrupt Priority Level (TOVF1 IPL)—Bits 15–14

This bit field is used to set the interrupt priority levels for certain peripheral IRQs. These IRQs are limited to priorities zero through two and are disabled by default.

- 00 = IRQ disabled (default)
- 01 = IRQ is priority level 0
- 10 = IRQ is priority level 1
- 11 = IRQ is priority level 2

8.7.7.2 Timer Channel 1 Compare Interrupt Priority Level (TCMP1 IPL)—Bits 13–12

This bit field is used to set the interrupt priority levels for certain peripheral IRQs. These IRQs are limited to priorities zero through two and are disabled by default.

- 00 = IRQ disabled (default)
- 01 = IRQ is priority level 0
- 10 = IRQ is priority level 1
- 11 = IRQ is priority level 2

8.7.7.3 Timer Channel 0 Input Edge Interrupt Priority Level (TINP0 IPL)—Bits 11–10

This bit field is used to set the interrupt priority levels for certain peripheral IRQs. These IRQs are limited to priorities zero through two and are disabled by default.

- 00 = IRQ disabled (default)
- 01 = IRQ is priority level 0
- 10 = IRQ is priority level 1
- 11 = IRQ is priority level 2

8.7.7.4 Timer Channel 0 Overflow Interrupt Priority Level (TOVF0 IPL)—Bits 9–8

This bit field is used to set the interrupt priority levels for certain peripheral IRQs. These IRQs are limited to priorities zero through two and are disabled by default.

- 00 = IRQ disabled (default)
- 01 = IRQ is priority level 0
- 10 = IRQ is priority level 1
- 11 = IRQ is priority level 2

8.7.7.5 Timer Channel 0 Compare Interrupt Priority Level (TCMP0 IPL)—Bits 7–6

This bit field is used to set the interrupt priority levels for certain peripheral IRQs. These IRQs are limited to priorities zero through two and are disabled by default.

- 00 = IRQ disabled (default)
- 01 = IRQ is priority level 0
- 10 = IRQ is priority level 1
- 11 = IRQ is priority level 2

8.7.7.6 Time of Day 1 Second Interval Interrupt Priority Level (TOD1 IPL)—Bits 5–4

This bit field is used to set the interrupt priority levels for certain peripheral IRQs. These IRQs are limited to priorities zero through two and are disabled by default.

- 00 = IRQ disabled (default)
- 01 = IRQ is priority level 0
- 10 = IRQ is priority level 1
- 11 = IRQ is priority level 2

8.7.7.7 Time of Day Alarm Interrupt Priority Level (TOD_ALRM IPL)—Bits 3–2

This bit field is used to set the interrupt priority levels for certain peripheral IRQs. These IRQs are limited to priorities zero through two and are disabled by default.

- 00 = IRQ disabled (default)
- 01 = IRQ is priority level 0
- 10 = IRQ is priority level 1
- 11 = IRQ is priority level 2

8.7.7.8 Host I/F Command Interrupt Priority Level (HOST_CMD IPL)—Bits 1–0

This bit field is used to set the interrupt priority levels for certain peripheral IRQs. These IRQs are limited to priorities zero through two and are disabled by default.

- 00 = IRQ disabled (default)
- 01 = IRQ is priority level 0
- 10 = IRQ is priority level 1
- 11 = IRQ is priority level 2

8.7.8 Interrupt Priority Register 7 (IPR7)

Base + $7	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	0	0	TINP3 IPL		TOVS3 IPL		TCMP3 IPL		TINP2 IPL		TOVF2 IPL		TCMP2 IPL		TINP1 IPL	
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 8-10. Interrupt Priority Register 7 (IPR7)

See Programmer's Sheet on Appendix page B - 33

8.7.8.1 Reserved—Bits 15–14

These bits are reserved or not implemented. They are read as 0, but they cannot be modified by writing.

8.7.8.2 Timer Channel 3 Input Edge Interrupt Priority Level (TINP3 IPL)—Bits 13–12

This bit field is used to set the interrupt priority levels for certain peripheral IRQs. These IRQs are limited to priorities zero through two and are disabled by default.

- 00 = IRQ disabled (default)
- 01 = IRQ is priority level 0
- 10 = IRQ is priority level 1
- 11 = IRQ is priority level 2

8.7.8.3 Timer Channel 3 Overflow Interrupt Priority Level (TOVF3 IPL)—Bits 11–10

This bit field is used to set the interrupt priority levels for certain peripheral IRQs. These IRQs are limited to priorities zero through two and are disabled by default.

- 00 = IRQ disabled (default)
- 01 = IRQ is priority level 0
- 10 = IRQ is priority level 1
- 11 = IRQ is priority level 2

8.7.8.4 Timer Channel 3 Compare Interrupt Priority Level (TCMP3 IPL)—Bits 9–8

This bit field is used to set the interrupt priority levels for certain peripheral IRQs. These IRQs are limited to priorities zero through two and are disabled by default.

- 00 = IRQ disabled (default)
- 01 = IRQ is priority level 0
- 10 = IRQ is priority level 1
- 11 = IRQ is priority level 2

8.7.8.5 Timer Channel 2 Input Edge Interrupt Priority Level (TINP2 IPL)—Bits 7–6

This bit field is used to set the interrupt priority levels for certain peripheral IRQs. These IRQs are limited to priorities zero through two and are disabled by default.

- 00 = IRQ disabled (default)
- 01 = IRQ is priority level 0
- 10 = IRQ is priority level 1
- 11 = IRQ is priority level 2

8.7.8.6 Timer Channel 2 Overflow Interrupt Priority Level (TOVF2 IPL)—Bits 5–4

This bit field is used to set the interrupt priority levels for certain peripheral IRQs. These IRQs are limited to priorities zero through two and are disabled by default.

- 00 = IRQ disabled (default)
- 01 = IRQ is priority level 0
- 10 = IRQ is priority level 1
- 11 = IRQ is priority level 2

8.7.8.7 Timer Channel 2 Compare Interrupt Priority Level (TCMP2 IPL)—Bits 3–2

This bit field is used to set the interrupt priority levels for certain peripheral IRQs. These IRQs are limited to priorities zero through two and are disabled by default.

- 00 = IRQ disabled (default)
- 01 = IRQ is priority level 0
- 10 = IRQ is priority level 1
- 11 = IRQ is priority level 2

8.7.8.8 Timer Channel 1 Input Edge Interrupt Priority Level (TINP1 IPL)—Bits 1–0

This bit field is used to set the interrupt priority levels for certain peripheral IRQs. These IRQs are limited to priorities zero through two and are disabled by default.

- 00 = IRQ disabled (default)
- 01 = IRQ is priority level 0
- 10 = IRQ is priority level 1
- 11 = IRQ is priority level 2

8.7.9 Interrupt Priority Register 8 (IPR8)

Base + $8	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	0	0	0	0	0	0	SCI1_RCV IPL		SCI1_RERR IPL		SCI1_RIDL IPL		SCI1_TIDL IPL		SCI1_XMIT IPL	
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 8-11. Interrupt Priority Register 8 (IPR8)

8.7.9.1 Reserved—Bits 15–10

These bits are reserved or not implemented. They are read as 0, but cannot be modified by writing.

8.7.9.2 Receiver Full Interrupt Priority Level (SCI1_RCV IPL)—Bits 9–8

These fields are used to set the interrupt priority levels for certain peripheral IRQs. These IRQs are limited to priorities zero through two and are disabled by default.

- 00 = IRQ disabled (default)
- 01 = IRQ is priority level 0
- 10 = IRQ is priority level 1
- 11 = IRQ is priority level 2

8.7.9.3 Receiver Error Interrupt Priority Level (SCI1_RERR IPL)—Bits 7–6

These fields are used to set the interrupt priority levels for certain peripheral IRQs. These IRQs are limited to priorities zero through two and are disabled by default.

- 00 = IRQ disabled (default)
- 01 = IRQ is priority level 0
- 10 = IRQ is priority level 1
- 11 = IRQ is priority level 2

8.7.9.4 Receiver Idle Interrupt Priority Level (SCI1_RIDL IPL)—Bits 5–4

These fields are used to set the interrupt priority levels for certain peripheral IRQs. These IRQs are limited to priorities zero through two and are disabled by default.

- 00 = IRQ disabled (default)
- 01 = IRQ is priority level 0
- 10 = IRQ is priority level 1
- 11 = IRQ is priority level 2

8.7.9.5 Transmitter Idle Interrupt Priority Level (SCI1_TIDL IPL)—Bits 3–2

These fields are used to set the interrupt priority levels for certain peripheral IRQs. These IRQs are limited to priorities zero through two and are disabled by default.

- 00 = IRQ disabled (default)
- 01 = IRQ is priority level 0
- 10 = IRQ is priority level 1
- 11 = IRQ is priority level 2

8.7.9.6 Transmitter Empty Interrupt Priority Level (SCI1_XMIT IPL) —Bits 1–0

These fields are used to set the interrupt priority levels for certain peripheral IRQs. These IRQs are limited to priorities zero through two and are disabled by default.

- 00 = IRQ disabled (default)
- 01 = IRQ is priority level 0
- 10 = IRQ is priority level 1
- 11 = IRQ is priority level 2

8.7.10 Vector Base Address Register (VBA)

Base + $9	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	0	0	0	VECTOR_BASE_ADDRESS												
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 8-12. Vector Base Address Register (VBA)

See Programmer's Sheet on Appendix page B - 37

8.7.10.1 Reserved—Bits 15–13

These bits are reserved or not implemented. They are read as 0, but they cannot be modified by writing.

8.7.10.2 Interrupt Vector Base Address (VECTOR_BASE_ADDRESS)—Bits 12–0

The value in this register is used as the upper 13 bits of the interrupt vector Vector Address Bus (VAB)[20:0]. The lower eight bits are determined based on the highest priority interrupt and are then appended onto this field (VBA) before presenting the full Vector Address Bus (VAB) to the 56800E core.

8.7.11 Fast Interrupt Match Registers (FIM0 and FIM1)

Base + $A	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	0	0	0	0	0	0	0	0	0	FAST INTERRUPT 0						
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 8-13. Fast Interrupt 0 Match Register (FIM0)

See Programmer's Sheet on Appendix page B - 38

Note: Although the 56800E core allows an unlimited number of fast interrupts, the 568xx family of chips limits this number to two in order to keep the size of the interrupt vector table small in terms of memory usage, and to reduce the overhead caused by each interrupt source having to provide a separate fast interrupt vector.

8.7.11.1 Reserved—Bits 15–7

These bits are reserved or not implemented. They are read as 0, but they cannot be modified by writing.

8.7.11.2 Fast Interrupt 0 Vector Number (FAST INTERRUPT 0)

These values are used to declare which two IRQs will be Fast Interrupts. Fast Interrupts vector directly to a service routine based on values in the Fast Interrupt Vector Address registers without having to go to a jump table first. IRQs used as Fast Interrupts *must* be set to priority Level 2. Unexpected results will occur if a Fast Interrupt vector is set to any other priority. Fast Interrupts automatically become the highest priority Level 2 Interrupt regardless of their location in the interrupt table prior to being declared as Fast Interrupt. Fast Interrupt 0 has priority over Fast Interrupt 1. To determine the vector number of each IRQ. Please refer to **Table 8-3.**

Base + $D	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	0	0	0	0	0	0	0	0	0	FAST INTERRUPT 1						
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 8-14. Fast Interrupt 1 Match Register (FIM1)

See Programmer's Sheet on Appendix page B - 39

8.7.11.3 Reserved—Bits 15–7

These bits are reserved or not implemented. They are read as 0, but they cannot be modified by writing.

8.7.11.4 Fast Interrupt 1 Vector Number (FAST INTERRUPT 1)

These values are used to declare which two IRQs will be Fast Interrupts. Fast Interrupts vector directly to a service routine based on values in the Fast Interrupt Vector Address registers without having to go to a jump table first. IRQs used as Fast Interrupts *must* be set to priority Level 2. Unexpected results will occur if a Fast Interrupt vector is set to any other priority. Fast Interrupts automatically become the highest priority Level 2 Interrupt regardless of their location in the interrupt table prior to being declared as Fast Interrupt. Fast Interrupt 0 has priority over Fast Interrupt 1. To determine the vector number of each IRQ. Please refer to **Table 8-3.**

8.7.12 Fast Interrupt Vector Address Registers (FIVAL0, FIVAH0, FIVAL1, FIVAH1)

The following four registers are combined to form two, 21-bit vector addresses for the Fast Interrupts defined in the FIM0 and FIM1 registers.

Base + $B	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	FAST INTERRUPT 0 VECTOR ADDRESS LOW															
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 8-15. Fast Interrupt 0 Vector Address Low Register (FIVAL0)

See Programmer's Sheet on Appendix page B - 40

8.7.12.1 Fast Interrupt 0 Vector Address Low—Bits 15–0

Lower 16 bits of Vector Address for Fast Interrupt 0.

Base + $C	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	0	0	0	0	0	0	0	0	0	0	0	FAST INTERRUPT 0 VECTOR ADDRESS HIGH				
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 8-16. Fast Interrupt 0 Vector Address High Register (FIVAH0)

See Programmer's Sheet on Appendix page B - 40

8.7.12.2 Reserved—Bits 15–5

These bits are reserved or not implemented. They are read as 0, but they cannot be modified by writing.

8.7.12.3 Fast Interrupt 0 Vector Address High—Bits 4–0

Upper five bits of Vector Address for Fast Interrupt 0.

8.7.13 Fast Interrupt 1 Vector Address Low Register (FIVAL1)

Base + $E	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	FAST INTERRUPT 1 VECTOR ADDRESS LOW															
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 8-17. Fast Interrupt 1 Vector Address Low Register (FIVAL1)

See Programmer's Sheet on Appendix page B - 41

8.7.13.1 Fast Interrupt 1 Vector Address Low—Bits 15–0

Lower 16 bits of Vector Address for Fast Interrupt 1

8.7.14 Fast Interrupt 1 Vector Address High Register (FIVAH1)

Base + $F	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	0	0	0	0	0	0	0	0	0	0	0	FAST INTERRUPT 1 VECTOR ADDRESS HIGH				
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 8-18. Fast Interrupt 1 Vector Address High Register (FIVAH1)

See Programmer's Sheet on Appendix page B - 41

8.7.14.1 Reserved—Bits 15–5

These bits are reserved or not implemented. They are read as 0, but they cannot be modified by writing.

8.7.14.2 Fast Interrupt 1 Vector Address Low—Bits 4–0

Upper five bits of Vector Address for Fast Interrupt 1.

8.7.15 IRQ Pending Registers (IRQP0, IRQP1, IRQP2, IRQP3, IRQP4)

These registers combine to represent the pending IRQs for interrupt vector numbers two through 69.

Base + $10	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	PENDING[16:1]															1
Write																
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Figure 8-19. IRQ Pending Register 0 (IRQP0)

See Programmer's Sheet on Appendix page B - 42

Base + $11	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	PENDING[32:17]															
Write																
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Figure 8-20. IRQ Pending Register 1 (IRQP1)

See Programmer's Sheet on Appendix page B - 42

Base + $12	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	PENDING[48:33]															
Write																
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Figure 8-21. IRQ Pending Register 2 (IRQP2)

See Programmer's Sheet on Appendix page B - 42

BAse + $13	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	PENDING[64:49]															
Write																
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Figure 8-22. IRQ Pending Register 3 (IRQP3)

See Programmer's Sheet on Appendix page B - 42

Base + $14	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	1	1	1	1	1	1	1	1	1	1	1	PENDING[69:65]				
Write																
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Figure 8-23. IRQ Pending Register 4 (IRQP4)

See Programmer's Sheet on Appendix page B - 42

8.7.15.1 Pending IRQs (PENDING)—Bits 2–69

- 0 = IRQ pending for this vector number
- 1 = No IRQ pending for this vector number

8.7.15.2 Reserved—Bits 70–80

These bits are reserved or not implemented. They are read as 1, but they cannot be modified by writing.

8.7.16 Interrupt Control Register (ICTL)

Base + $1A	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	INT	IPIC		VAB							INT_DIS	0	IRQB STATE	IRQA STATE	IRQB EDG	IRQA EDG
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0

Figure 8-24. Interrupt Control Register (ICTL)

See Programmer's Sheet on Appendix page B - 43

8.7.16.1 Interrupt (INT)—Bit 15

This bit reflects the state of the interrupt to the 56800E core.

- 0 = No interrupt is being sent to the core
- 1 = An interrupt is being sent to the core

8.7.16.2 Interrupt Priority Level (IPIC)—Bits 14–13

These bits reflect the state of the new interrupt priority level bits being presented to the 56800E core at the time the last IRQ was taken. This field is only updated when the 56800E core jumps to a new interrupt service routine.

Note: Nested interrupts may cause this field to be updated before the original interrupt service routine can read it.

- 00 = Required nested exception priority levels are 0, 1, 2, or 3

- 01 = Required nested exception priority levels are 1, 2, or 3
- 10 = Required nested exception priority levels are 2 or 3
- 11 = Required nested exception priority level is 3

8.7.16.3 Vector Number (VAB)—Bits 12–6

This bit field shows the Vector Number, described in **Table 8-3** of the last IRQ taken. This bit field is only updated when the 56800E core jumps to a new interrupt service routine.

Note: Nested interrupts may cause this field to be updated before the original interrupt service routine can read it.

8.7.16.4 Interrupt Disable (INT_DIS)—Bit 5

This bit allows the user to disable all interrupts.

- 0 = Normal operation (default)
- 1 = All interrupts disabled

8.7.16.5 Reserved—Bit 4

This bit is reserved or not implemented. It is read as 0, but they cannot be modified by writing.

8.7.16.6 State of $\overline{\text{IRQB}}$ (IRQB STATE)—Bit 3

This bit reflects the state of the external $\overline{\text{IRQB}}$ pin.

8.7.16.7 State of $\overline{\text{IRQA}}$ (IRQA STATE)—Bit 2

This bit reflects the state of the external $\overline{\text{IRQA}}$ pin.

8.7.16.8 IRQB Edge (IRQB EDG)—Bit 1

This bit controls whether the external $\overline{\text{IRQB}}$ interrupt is edge or level sensitive. During Stop and Wait modes it is automatically level sensitive.

- 0 = $\overline{\text{IRQB}}$ interrupt is low level sensitive (default)
- 1 = $\overline{\text{IRQB}}$ interrupt is falling edge sensitive

8.7.16.9 IRQA Edge (IRQA EDG)—Bit 0

This bit controls whether the external $\overline{\text{IRQA}}$ interrupt is edge or level sensitive. During Stop and Wait modes it is automatically level sensitive.

- 0 = $\overline{\text{IRQA}}$ interrupt is low level sensitive (default)
- 1 = $\overline{\text{IRQA}}$ interrupt is falling edge sensitive

8.7.17 Interrupt Vector Map

Table 8-3 provides the list of interrupt vectors on the 56853 through 858 devices. As noted in the table, the total vector table size is 128 vectors or 256 words of memory. This table also provides the allowable priority range or fixed priority for each IRQ.

Table 8-3. Interrupt Vector Table Contents

Peripheral	Vector Number	Priority Level	Vector Base Address +	Interrupt Function	Chip Exceptions
Core	0	3	P:$00	Reserved	
Core	1	3	P:$02	Reserved	
Core	2	3	P:$04	Illegal Instruction	—
Core	3	3	P:$06	SW Interrupt 3	—
Core	4	3	P:$08	HW Stack Overflow	—
Core	5	3	P:$0A	Misaligned Long Word Access	—
Core	6	1-3	P:$0C	EOnCE Step Counter	—
Core	7	1-3	P:$0E	EOnCE Breakpoint Unit 0	—
Core	8	1-3	P:$10	Reserved	
Core	9	1-3	P:$12	EOnCE Trace Buffer	—
Core	10	1-3	P:$14	EOnCE Transmit Register Empty	—
Core	11	1-3	P:$16	EOnCE Receive Register Full	—
Core	12	0-3	P:$18	Reserved	
Core	13	0-3	P:$1A	Reserved	
Core	14	2	P:$1C	SW Interrupt 2	—
Core	15	1	P:$1E	SW Interrupt 1	—
Core	16	0	P:$20	SW Interrupt 0	—
Core	17	0-2	P:$22	IRQA	—
Core	18	0-2	P:$24	IRQB	—
Core	19	0-2	P:$26	Reserved	
PLL	20	0-2	P:$28	PLL Loss Of Lock	—
—	21	0-2	P:$2A	Reserved	
DMA	22	0-2	P:$2C	DMA_DONE 0	—
DMA	23	0-2	P:$2E	DMA_DONE 1	—
DMA	24	0-2	P:$30	DMA_DONE 2	—
DMA	25	0-2	P:$32	DMA_DONE 3	—
DMA	26	0-2	P:$34	DMA_DONE 4	—
DMA	27	0-2	P:$36	DMA_DONE 5	—
ESSI 0	28	0-2	P:$38	ESSI 0 Receive Data with Exception Status	—
ESSI 0	29	0-2	P:$3A	ESSI 0 Receive Data	—
ESSI 0	30	0-2	P:$3C	ESSI 0 Receive Last Slot	—
ESSI 0	31	0-2	P:$3E	ESSI 0 Transmit Data with Exception Status	—

Table 8-3. Interrupt Vector Table Contents (Continued)

Peripheral	Vector Number	Priority Level	Vector Base Address +	Interrupt Function	Chip Exceptions
ESSI 0	32	0-2	P:$40	ESSI 0 Transmit Data	—
ESSI 0	33	0-2	P:$42	ESSI 0 Transmit Last Slot	—
ESSI 1	34	0-2	P:$44	ESSI 1 Receive Data w/ Exception Status	Only for 857-858
ESSI 1	35	0-2	P:$46	ESSI 1 Receive Data	Only for 857-858
ESSI 1	36	0-2	P:$48	ESSI 1 Receive Last Slot	Only for 857-858
ESSI 1	37	0-2	P:$4A	ESSI 1 Transmit Data w/ Exception Status	Only for 857-858
ESSI 1	38	0-2	P:$4C	ESSI 1 Transmit Data	Only for 857-858
ESSI 1	39	0-2	P:$4E	ESSI 1 Transmit Last Slot	Only for 857-858
SPI	40	0-2	P:$50	SPI Receiver Full	Not on 855
SPI	41	0-2	P:$52	SPI Transmitter Empty	Not on 855
SCI 0	42	0-2	P:$54	SCI 0 Transmitter Empty	—
SCI 0	43	0-2	P:$56	SCI 0 Transmitter Idle	—
SCI 0	44	0-2	P:$58	SCI 0 Receiver Idle	—
SCI 0	45	0-2	P:$5A	SCI 0 Receiver Error (Receiver Overrun, Noise Error, Framing Error, Parity Error)	—
SCI 0	46	0-2	P:$5C	SCI 0 Receiver Full	—
HI8	47	0-2	P:$5E	Host Receive Data	Not on 855
HI8	48	0-2	P:$60	Host Transmit Data	Not on 855
HI8	49	0-2	P:$62	Host Command (default)	Not on 855
TOD	50	0-2	P:$64	TOD Alarm	—
TOD	51	0-2	P:$66	TOD One Second Interval	—
Timer	52	0-2	P:$68	Timer Compare 0	—
Timer	53	0-2	P:$6A	Timer Overflow 0	—
Timer	54	0-2	P:$6C	Timer Input Edge Flag 0	—
Timer	55	0-2	P:$6E	Timer Compare 1	—
Timer	56	0-2	P:$70	Timer Overflow 1	—
Timer	57	0-2	P:$72	Timer Input Edge Flag 1	—
Timer	58	0-2	P:$74	Timer Compare 2	—
Timer	59	0-2	P:$76	Timer Overflow 2	—
Timer	60	0-2	P:$78	Timer Input Edge Flag 2	—
Timer	61	0-2	P:$7A	Timer Compare 3	—
Timer	62	0-2	P:$7C	Timer Overflow 3	—
Timer	63	0-2	P:$7E	Timer Input Edge Flag 3	—
core	64	-1	P:$80	SW interrupt LP	—
SCI 1	65	0-2	P:$82	SCI 1 Transmitter Empty	—
SCI 1	66	0-2	P:$84	SCI 1 Transmitter Idle	—
SCI 1	67	0-2	P:$86	SCI 1 Receiver Idle	—
SCI 1	68	0-2	P:$88	SCI 1 Receiver Error (Receiver Overrun, Noise Error, Framing Error, Parity Error)	—

Table 8-3. Interrupt Vector Table Contents (Continued)

Peripheral	Vector Number	Priority Level	Vector Base Address +	Interrupt Function	Chip Exceptions
SCI 1	69	0-2	P:$8A	SCI 1 Receiver Full	—
HI8	70		P:$8C	Available for Host Command	Not on 855
HI8	71		P:$8E	Available for Host Command	Not on 855
HI8	72		P:$90	Available for Host Command	Not on 855
HI8	73		P:$92	Available for Host Command	Not on 855
HI8	74	**Note:** Controlled by Vector 49	P:$94	Available for Host Command	Not on 855
HI8	75		P:$96	Available for Host Command	Not on 855
HI8	76		P:$98	Available for Host Command	Not on 855
HI8	77		P:$9A	Available for Host Command	Not on 855
HI8	78		P:$9C	Available for Host Command	Not on 855
HI8	79		P:$9E	Available for Host Command	Not on 855
HI8	80		P:$A0	Available for Host Command	Not on 855
HI8	81		P:$A2	Available for Host Command	Not on 855
HI8	82		P:$A4	Available for Host Command	Not on 855
HI8	83		P:$A6	Available for Host Command	Not on 855
HI8	84		P:$A8	Available for Host Command	Not on 855
HI8	85		P:$AA	Available for Host Command	Not on 855
HI8	86		P:$AC	Available for Host Command	Not on 855
HI8	87		P:$AE	Available for Host Command	Not on 855
HI8	88		P:$B0	Available for Host Command	Not on 855
HI8	89		P:$B2	Available for Host Command	Not on 855
HI8	90		P:$B4	Available for Host Command	Not on 855
HI8	91		P:$B6	Available for Host Command	Not on 855
HI8	92		P:$B8	Available for Host Command	Not on 855
HI8	93		P:$BA	Available for Host Command	Not on 855
HI8	94		P:$BC	Available for Host Command	Not on 855
HI8	95		P:$BE	Available for Host Command	Not on 855
HI8	96		P:$C0	Available for Host Command	Not on 855
HI8	97		P:$C2	Available for Host Command	Not on 855
HI8	98		P:$C4	Available for Host Command	Not on 855
HI8	99		P:$C6	Available for Host Command	Not on 855
HI8	100		P:$C8	Available for Host Command	Not on 855
HI8	101		P:$CA	Available for Host Command	Not on 855
HI8	102		P:$CC	Available for Host Command	Not on 855
HI8	103		P:$CE	Available for Host Command	Not on 855
HI8	104		P:$D0	Available for Host Command	Not on 855
HI8	105		P:$D2	Available for Host Command	Not on 855
HI8	106		P:$D4	Available for Host Command	Not on 855
HI8	107		P:$D6	Available for Host Command	Not on 855
HI8	108		P:$D8	Available for Host Command	Not on 855
HI8	109		P:$DA	Available for Host Command	Not on 855
HI8	110		P:$DC	Available for Host Command	Not on 855

Table 8-3. Interrupt Vector Table Contents (Continued)

Peripheral	Vector Number	Priority Level	Vector Base Address +	Interrupt Function	Chip Exceptions
HI8	111		P:$DE	Available for Host Command	Not on 855
HI8	112	**Note:** Controlled by Vector 49	P:$E0	Available for Host Command	Not on 855
HI8	113		P:$E2	Available for Host Command	Not on 855
HI8	114		P:$E4	Available for Host Command	Not on 855
HI8	115		P:$E6	Available for Host Command	Not on 855
HI8	116		P:$E8	Available for Host Command	Not on 855
HI8	117		P:$EA	Available for Host Command	Not on 855
HI8	118		P:$EC	Available for Host Command	Not on 855
HI8	119		P:$EE	Available for Host Command	Not on 855
HI8	120		P:$F0	Available for Host Command	Not on 855
HI8	121		P:$F2	Available for Host Command	Not on 855
HI8	122		P:$F4	Available for Host Command	Not on 855
HI8	123		P:$F6	Available for Host Command	Not on 855
HI8	124		P:$F8	Available for Host Command	Not on 855
HI8	125		P:$FA	Available for Host Command	Not on 855
HI8	126		P:$FC	Available for Host Command	Not on 855
HI8	127		P:$FE	Available for Host Command	Not on 855

8.8 Wait and Stop Mode Operations

The system clocks and the 56800E are turned off during Wait and Stop modes. The ITCN will signal a pending IRQ to the System Integration Module (SIM) to restart the clocks and service the IRQ. An IRQ can only wake up the core if the IRQ is enabled prior to entering the Wait or Stop mode. Also, the $\overline{\text{IRQA}}$ and $\overline{\text{IRQB}}$ signals automatically become low level sensitive in these modes even if the Control Register bits are set to make them falling edge sensitive. This is because there is no clock available to detect the falling edge.

8.9 Host Control Interrupt Vector

The Host Command IRQ, vector 49, acts differently than the other IRQs. The vector 49 value driven onto the lower eight bits of VAB in response to this IRQ isn't fixed as it is for the other IRQs. The Host I/F module supplies these lower bits and the ITCN uses them when the interrupt to the 56800E is in response to the Host Command IRQ. The ITCN also asserts the HP_IRQ_ACK signal back to the Host Port module when the ITCN receives acknowledgement from the CPU in response to the Host Command IRQ.

8.10 Resets

8.10.1 Reset Handshake Timing

The ITCN provides the 56800E with a reset vector address on the VAB pins whenever $\overline{\text{RST}}$ is asserted. The Reset Vector will be presented until the first rising clock edge after $\overline{\text{RST}}$ is released. The general timing is shown in **Figure 8-25.**

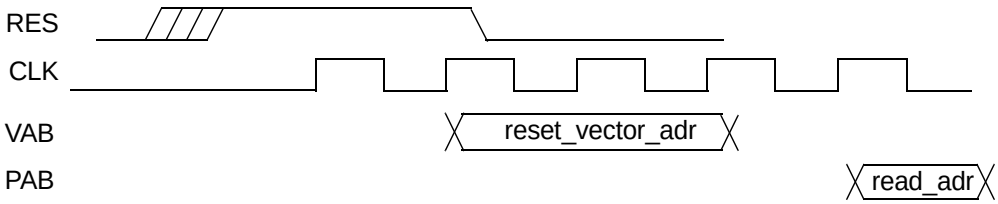


Figure 8-25. Reset Interface

8.10.2 ITCN After Reset

After reset, all of the ITCN registers are in their default states. This means all interrupts are disabled except the core IRQs with fixed priorities. Those are:

- Illegal Instruction
- SW Interrupt 3
- HW Stack Overflow
- Misaligned Long Word Access
- SW Interrupt 2
- SW Interrupt 1
- SW Interrupt 0
- SW Interrupt LP

Core IRQs are enabled at their fixed priority levels.

8.11 Interrupts

8.11.1 Interrupt Handshake Timing

The control logic looks a the current interrupt processing level using the SR_REG[9:8] bits from the 56800E core and determines if an interrupt request of sufficient priority exists to assert the interrupt output to the core. Upon asserting INT to the core, the Interrupt Controller also asserts new values for the IPIC_LEVEL pins. These pins indicate the priority level required to interrupt

this newly requested interrupt. The core will latch IPIC_LEVEL and it will be driven back to the Interrupt Controller as new values on the SR_REG[9:8] pins.

When the 56800E core recognizes the assertion of the interrupt pin, it will deassert PIC_EN which tells the Interrupt Controller to drive VAB with the address corresponding to the highest priority interrupt request in order to start the interrupt service routine. When the 56800E core asserts the IACK signal, the Interrupt Controller will deassert the interrupt signal to the core. The controller will not reassert the interrupt signal until PIC_EN is asserted by the 56800E core.

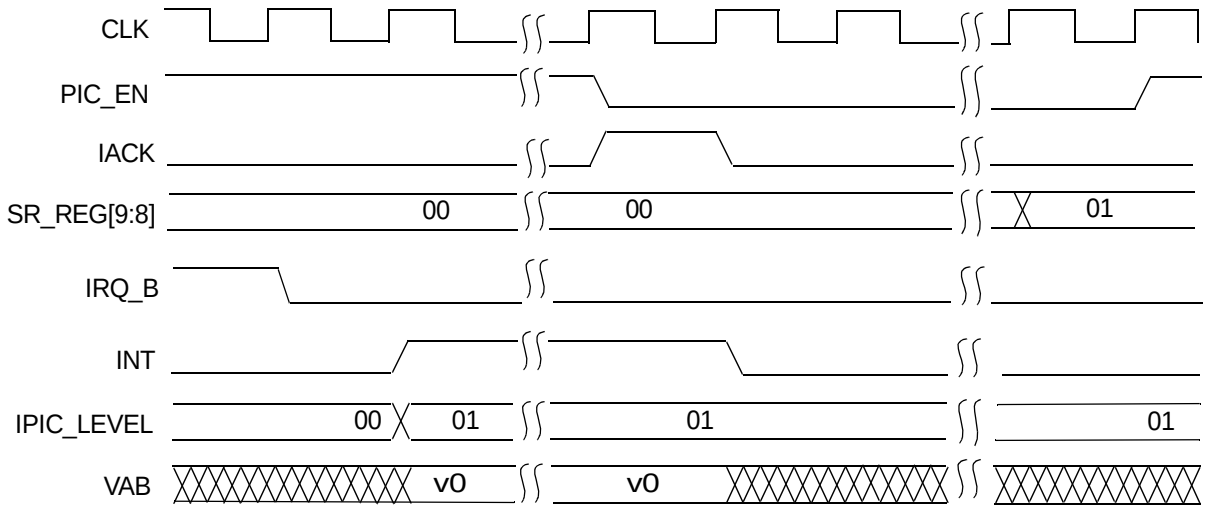


Figure 8-26. Interrupt Handshake Timing

8.11.2 Interrupt Nesting

Interrupt exceptions may be nested to allow an IRQ of higher priority than the current exception to be serviced. The following tables define the nesting requirements for each priority level.

Table 8-4. Interrupt Mask Bit Definition

SR	SR	Exceptions Permitted	Exceptions Masked
0	0	Priorities 0, 1, 2, 3	None
0	1	Priorities 1, 2, 3	Priority 0
1	0	Priorities 2,3	Priorities 0, 1
1	1	Priority 3	Priorities 0, 1, 2

Table 8-5. Interrupt Priority Encoding

IPIC_LEVEL	Current Interrupt Priority Level	Required Nested Exception Priority
00	No interrupt or SWILP	Priorities 0, 1, 2, 3
01	Priority 0	Priorities 1, 2, 3
10	Priority 1	Priorities 2, 3
11	Priorities 2 or 3	Priority 3

Chapter 9
Direct Memory Access (DMA)

9.1 Introduction

The Direct Memory Access (DMA) controller transfers data between points in the memory map without intervention by the CPU. The DMA controller allows movements of data to and from internal data memory or internal peripherals such as SSI, SCI, SPI, and so on, to occur in the background of CPU operation. The 5685x packages include six independent programmable DMA channels, allowing six different contexts for DMA operation.

9.2 Features

The DMA features consist of:

- DMA perpetration independently of the CPU
- Six channels supporting six independent block transfers
- The DMA Controller has access to a maximum of one-third of the system bus traffic. This duty cycle is governed by the 5685x System Bus Controller.
- Each channel has independently programmable peripheral selection.
- Each channel's source and destination address registers indices can be configured. For each memory *fetch* and *put*, the address may be post-incriminated, post-decremented, or remain constant.
- Each read/write transfer may be initiated by selected events, specified peripheral requests, or direct CPU triggering may launch a new DMA transaction.
- Upon completion of a block transfer, each DMA channel may send an interrupt to the CPU.
- The DMA includes a *circular queue* operational mode to provide continuous DMA operation with no additional processor intervention.

9.3 Signal Description

The DMA controller signals are described in **Table 9-1** through **Table 9-3.**

Table 9-1. DMA Common Signals

Name	Type	Function	Clock Domain
$\overline{\text{RESET}}$	Input	DMA Reset clears all internal registers and resets DMA state machine.	—

Table 9-2. DMA IPBus Signals

Name	Type	Function	Clock Domain
IPBUS_CLK	Input	IPBus Clock	—
IPBUS_ADDR[2:0]	Input	DMA IPBus Register Select	IPBUS_CLK
IPBUS_DATA_IN[15:0]	Input	Write data from IPBus to DMA registers	IPBUS_CLK
IPBUS_DATA_OUT[15:0]	Output	Read data to IPBus from DMA registers	IPBUS_CLK
IPBUS_R/W	Input	IPBus read/write signal	IPBUS_CLK
IPBUS_SEL	Input	Select signal for DMA Controller	IPBUS_CLK
IPBUS_REQ[30:0]	Input	DMA request signal from IPBus peripheral	IPBUS_CLK
DMA_$\overline{\text{INTR}}$	Output	DMA interrupt request signal (active low)	IPBUS_CLK
IP_PHASE	Input	Data signal indicating IPBus phase	SYS_CLK

Table 9-3. DMA X1 Bus Signals

Name	Type	Function	Clock Domain
SYS_CLK	Input	System Clock	—
X1_BUS_REQUEST	Output	DMA request to System Bus Controller (SBC)	SYS_CLK
X1_BUS_GRANT	Input	DMA grant signal from SBC	SYS_CLK
X1_ADDR_OUT[23:0]	Output	DMA-provided data space address	SYS_CLK
X1_W_DATA[31:0]	Output	DMA-provided data	SYS_CLK
X1_W	Output	DMA system bus write signal	SYS_CLK
X1_R_DATA[31:0]	Input	SBC-provided system bus read data	SYS_CLK
X1_R	Output	DMA system bus read signal	SYS_CLK
X1_SIZE	Output	Size of data written (byte, word, long)	SYS_CLK

9.4 Module Memory Maps

The 5685x devices contain six independent DMA channels. Each DMA channel includes seven user-programmable 16-bit registers. Base addresses of each register set are provided in **Table 9-4** through **Table 9-9.**

Table 9-4. DMA 0 Register Address Map (DMA0_BASE = $1FFEC0)

Address Offset	Register Acronym	Register Name	Accesss Type	Chapter Location
Base + $0	DMA_3_TC	Transfer Control	Read/Write	Section 9.6.1
Base + $1	DMA_3_CQS	Circular Queue Size	Read/Write	Section 9.6.2
Base + $2	DMA_3_TC	Transfer Count	Read/Write	Section 9.6.3
Base + $3	DMA_3_DAL	Destination Address-Low	Read/Write	Section 9.6.4
Base + $4	DMA_3_DAH	Destination Address-High	Read/Write	Section 9.6.5
Base + $5	DMA_3_SAL	Source Address-Low	Read/Write	Section 9.6.6
Base + $6	DMA_3_SAH	Source Address-High	Read/Write	Section 9.6.7

Table 9-5. DMA 1 Register Address Map (DMA1_BASE = $1FFEC8)

Address Offset	Register Acronym	Register Name	Accesss Type	Chapter Location
Base + $0	DMA_3_TC	Transfer Control	Read/Write	Section 9.6.1
Base + $1	DMA_3_CQS	Circular Queue Size	Read/Write	Section 9.6.2
Base + $2	DMA_3_TC	Transfer Count	Read/Write	Section 9.6.3
Base + $3	DMA_3_DAL	Destination Address-Low	Read/Write	Section 9.6.4
Base + $4	DMA_3_DAH	Destination Address-High	Read/Write	Section 9.6.5
Base + $5	DMA_3_SAL	Source Address-Low	Read/Write	Section 9.6.6
Base + $6	DMA_3_SAH	Source Address-High	Read/Write	Section 9.6.7

Table 9-6. DMA 2 Register Address Map (DMA2_BASE = $1FFED0)

Address Offset	Register Acronym	Register Name	Access Type	Chapter Location
Base + $0	DMA_3_TC	Transfer Control	Read/Write	Section 9.6.1
Base + $1	DMA_3_CQS	Circular Queue Size	Read/Write	Section 9.6.2
Base + $2	DMA_3_TC	Transfer Count	Read/Write	Section 9.6.3
Base + $3	DMA_3_DAL	Destination Address-Low	Read/Write	Section 9.6.4
Base + $4	DMA_3_DAH	Destination Address-High	Read/Write	Section 9.6.5
Base + $5	DMA_3_SAL	Source Address-Low	Read/Write	Section 9.6.6
Base + $6	DMA_3_SAH	Source Address-High	Read/Write	Section 9.6.7

Table 9-7. DMA 3 Register Address Map (DMA3_BASE = $1FFED8)

Address Offset	Register Acronym	Register Name	Access Type	Chapter Location
Base + $0	DMA_3_TC	Transfer Control	Read/Write	Section 9.6.1
Base + $1	DMA_3_CQS	Circular Queue Size	Read/Write	Section 9.6.2
Base + $2	DMA_3_TC	Transfer Count	Read/Write	Section 9.6.3
Base + $3	DMA_3_DAL	Destination Address-Low	Read/Write	Section 9.6.4
Base + $4	DMA_3_DAH	Destination Address-High	Read/Write	Section 9.6.5
Base + $5	DMA_3_SAL	Source Address-Low	Read/Write	Section 9.6.6
Base + $6	DMA_3_SAH	Source Address-High	Read/Write	Section 9.6.7

Table 9-8. DMA 4 Register Address Map (DMA4_BASE = $1FFEE0)

Address Offset	Register Acronym	Register Name	Access Type	Chapter Location
Base + $0	DMA_4_TC	Transfer Control	Read/Write	Section 9.6.1
Base + $1	DMA_4_CQS	Circular Queue Size	Read/Write	Section 9.6.2
Base + $2	DMA_4_TC	Transfer Count	Read/Write	Section 9.6.3
Base + $3	DMA_4_DAL	Destination Address-Low	Read/Write	Section 9.6.4
Base + $4	DMA_4_DAH	Destination Address-High	Read/Write	Section 9.6.5
Base + $5	DMA_4_SAL	Source Address-Low	Read/Write	Section 9.6.6
Base + $6	DMA_4_SAH	Source Address-High	Read/Write	Section 9.6.7

Table 9-9. DMA 5 Register Address Map (DMA5_BASE = $1FFEE8)

Address Offset	Register Acronym	Register Name	Access Type	Chapter Location
Base + $0	DMA_5_TC	Transfer Control	Read/Write	Section 9.6.1
Base + $1	DMA_5_CQS	Circular Queue Size	Read/Write	Section 9.6.2
Base + $2	DMA_5_TC	Transfer Count	Read/Write	Section 9.6.3
Base + $3	DMA_5_DAL	Destination Address-Low	Read/Write	Section 9.6.4
Base + $4	DMA_5_DAH	Destination Address-High	Read/Write	Section 9.6.5
Base + $5	DMA_5_SAL	Source Address-Low	Read/Write	Section 9.6.6
Base + $6	DMA_5_SAH	Source Address-High	Read/Write	Section 9.6.7

Add. Offset	Register Name		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
$0	DMATC	R	DMA ON	INTR ON	INTR PEND	DATA SIZE	0	0	0	PERIPH SEL					SS/M		DS/M	
		W					0	0	0									
$1	DMACQS	R	DMA CIRCULAR QUEUE SIZE															
		W																
$2	DMACNT	R	DMA TRANSFER COUNT															
		W																
$3	DAMDAL	R	DMA DESTINATION ADDRESS (LOW)															
		W																
$4	DMADAH	R	0	0	0	0	0	0	0	0	DMA DESTINATION ADDRESS (HIGH)							
		R	0	0	0	0	0	0	0	0								
$5	DMASAL	R	DMA SOURCE ADDRESS (LOW)															
		W																
$6	DMASAH	R	0	0	0	0	0	0	0	0	DMA SOURCE ADDRESS (HIGH)							
		W	0	0	0	0	0	0	0	0								

R	0	Read as 0
W		Reserved

Figure 9-1. DMA Register Map Summary

9.5 DMA Controller Block Diagram

One channel of the DMA Controller is depicted in **Figure 9-2.**

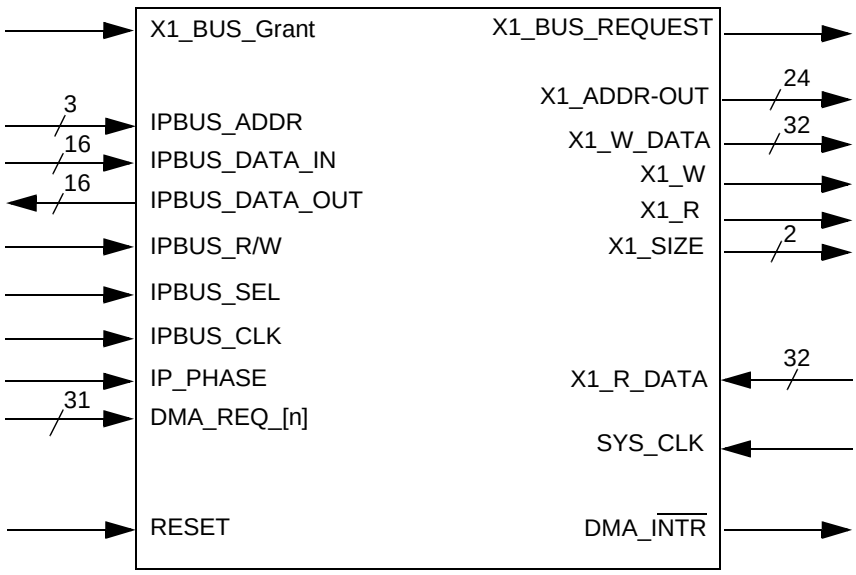


Figure 9-2. DMA Controller

9.6 Register Descriptions (DMA_BASES = $1FFEC0, $1FFEC8, $1FFED0, $1FFED8, $1FFEE0, $1FFEE8)

9.6.1 DMA Transfer Control (DMATC)

Base + $0	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	DMAON	INTRON	INTR_PEND	DATA_ SIZE	0	0	0	PERIPH SEL					SS/M		DS/M	
Write					0	0	0									
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 9-3. DMA Transfer Control (DMATC)

See Programmer's Sheet on Appendix page B - 51

9.6.1.1 Direct Memory Access On (DMAON)—Bit 15

This bit enables/disables DMA operation. When the Circular Queue operation is disabled and the DMACQS register is zero, this bit is automatically cleared at the conclusion of a DMA block transfer or when the Transfer Count register decrements to zero.

- 0 = DMA Operation is disabled
- 1 = DMA Operation is enabled

9.6.1.2 Interrupt On (INTRON)—Bit 14

This bit enables/disables DMA interrupt operation. If DMA interrupt operation is enabled, the DMA controller issues an interrupt whenever the DMA Transfer Count Register (DMACNT) decrements to zero.

- 0 = DMA Interrupt Operation is disabled
- 1 = DMA Interrupt Operation is enabled

9.6.1.3 Interrupt Pending (INTRPEND)—Bit 13

This bit is the DMA Interrupt Pending bit. When this bit is asserted, a DMA interrupt is pending; when this bit is deasserted, no DMA interrupt is pending.

- 0 = A DMA Interrupt is not pending
- 1 = A DMA Interrupt is pending

9.6.1.4 Data Size (DATASIZE)—Bit 12

This bit defines the data size of the DMA transfers and determines whether the DMA address registers point to words or bytes.

- 0 = DMA transfers are performed in bytes (8 bits), with address registers pointing to bytes.
- 1 = DMA transfers are performed in words (16 bits), with address registers pointing to words.

9.6.1.5 Reserved—Bits 11–9

These bits are reserved or not implemented. They are read as, and written with 0s.

9.6.1.6 Peripheral Select (PERIPH_SEL)—Bits 8–4

This five-bit field selects one of 16 possible DMA peripherals. **Table 9-10** illustrates the peripheral selected for each setting of this field. When the DMA is enabled, the peripheral chosen by this field controls the start of the DMA operation. That is, when the selected DMA_REQ [PERIPHERAL SELECT] line is asserted, the DMA operation begins.

Note: If PERIPHERAL SELECT is equal to zero, DMA operation begins immediately after DMA operation is enabled (via asserting the DMA ON bit). This mechanism may be used for memory-to-memory transfers. If a peripheral-to-peripheral DMA transfer is desired, PERIPHERAL SELECT should be set to the *slower* of the two peripherals to prevent data overruns.

Note: Whenever doing DMA transfers with peripherals, DATASIZE should be set as word mode.

Table 9-10. DMA_REQ Connections

DMA_REQ Line	Peripheral	Transfer Control Register Peripheral Select Field
15	HI8 High Transmitter	16
14	HI8 High Receiver	15
13	ESSI1 Transmitter 2	14
12	ESSI1 Transmitter 1	13
11	ESSI1 Transmitter 0	12
10	ESSI1 Receiver	11
9	ESSI0 Transmitter 2	10
8	ESSI0 Transmitter 1	9
7	ESSI0 Transmitter 0	8
6	ESSI0 Receiver	7
5	SCI1 Transmitter	6
4	SCI1 Receiver	5
3	SCI0 Transmitter	4
2	SCI0 Receiver	3
1	SPI Transmitter	2
0	SPI Receiver	1

To select a peripheral to trigger a DMA operation, the peripheral select field of the DMA Transfer Control Register is written with the value *DMA_REQ line* + 1. For example, if you want the SCI1 Receiver to initiate a DMA request, write the value 5 into the peripheral select field of the DMA Transfer Control Register.

9.6.1.7 Source Sign/Magnitude (SS/M)—Bits 3–2

This two-bit field determines how the DMA Source Address Register gets changed after each DMA read operation. The Source Address Register may remain unchanged, be post-incriminated, or be post-decremented as follows:

- 0x = Do not change the address register
- 10 = Increment the address register (by the DATA SIZE amount) after each read
- 11 = Decrement the address register (by the DATA SIZE amount) after each read

9.6.1.8 Destination Sign/Magnitude (DS/M)—Bits 1–0

This two-bit field determines how the DMA Destination Address Register gets changed after each DMA write operation. The Destination Address Register may remain unchanged, be post-incriminated, or be post-decremented as follows:

- 0x = Do not change the address register
- 10 = Increment the address register (by the DATA SIZE amount) after each read
- 11 = Decrement the address register (by the DATA SIZE amount) after each read

9.6.2 DMA Circular Queue Size (DMACQS)

Base+ $1	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	DMA CIRCULAR QUEUE SIZE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 9-4. DMA Circular Queue Size (DMACQS)

See Programmer's Sheet on Appendix page B - 50

9.6.2.1 Circular Queue Size (DMACQS)—Bits 15–0

This register determines the size of the Circular Queue. It defaults to all zeros. When zero, circular queue operation is disabled. When the Circular Queue Size is non-zero, circular queue mode is operational. In this mode, the DMA Destination Address Register changes with post-increments or post-decrements for each DMA write operation. The Circular Queue Size is decremented after each write. When the Circular Queue Size reaches zero it is restored to its initial value. The DMA Destination Address Register is also restored to its initial value. This

restoration allows the circular queue to continuously update with new data without additional Central Processing Unit (CPU) intervention.

Note: If interrupts are enabled while in the Circular Queue mode the DMA Transfer Count Register is used to determine when interrupts are issued.

Note: If *both* the source *and* destination registers are set to change using post- increment or post-decrement, *both* registers will be returned to their initial values.

9.6.3 DMA Transfer Count (DMACNT)

Base + $2	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	DMA TRANSFER COUNT															
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 9-5. DMA Transfer Count (DMACNT)

See Programmer's Sheet on Appendix page B - 49

9.6.3.1 Transfer Count (DMACNT)—Bits 15–0

This is the count of the number of items (bytes or words) to transfer in the DMA data block. This value is decremented after each DMA write operation. If DMA interrupts are enabled, a DMA interrupt will be asserted when this count reaches zero. If circular queue operation is enabled, the Transfer Count register is reloaded with its initial count value when it reaches zero. This permits continuous DMA operation without additional Central Processing Unit (CPU) intervention. In the circular queue operational mode and interrupts enabled an interrupt is issued each time the transfer count reaches zero, if DMA. Each interrupt indicates the complete transfer of one count-sized block of data.

9.6.4 DMA Destination Address Low (DMADAL)

Base+ $3	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	DMA DESTINATION ADDRESS (LOW)															
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 9-6. DMA Destination Address Low (DMADAL)

See Programmer's Sheet on Appendix page B - 48

9.6.4.1 Destination Address Low (DMADAL)—Bits 15–0

This is the lower word of the DMA Destination Address. DMA data is written from this address. If post-incrementing (or decrementing) is enabled, this value is updated with each write transaction.

9.6.5 DMA Destination Side Address High (DMADAH)

Base + $4	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	0	0	0	0	0	0	0	0	DMA DESTINATION ADDRESS (HIGH)							
Write	0	0	0	0	0	0	0	0								
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 9-7. DMA Destination Address High (DMADAH)

See Programmer's Sheet on Appendix page B - 47

9.6.5.1 Reserved—Bits 15–8

These bits are reserved or not implemented. They are read as, and written with 0s.

9.6.5.2 Destination Address High (DMADAH)—Bits 7–0

This is the upper byte of the DMA Destination Address. DMA data is written from this address. If post-incrementing (or decrementing) is enabled, this value is updated with each write transaction.

9.6.6 DMA Source Address Low (DMASAL)

Base + $5	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	DMA SOURCE ADDRESS (LOW)															
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 9-8. DMA Source Address Low (DMASAL)

See Programmer's Sheet on Appendix page B - 46

9.6.6.1 Source Address Low (DMASAL)—Bits 15-0

This is the lower word of the DMA Source Address. DMA data is read from this address. If post-incrementing (or decrementing) is enabled, this value is updated with each read transaction.

9.6.7 DMA Source Address High (DMASAH)

Base + $6	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	0	0	0	0	0	0	0	0	DMA SOURCE ADDRESS (HIGH)							
Write	0	0	0	0	0	0	0	0								
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 9-9. DMA Source Address High (DMASAH)

See Programmer's Sheet on Appendix page B - 45

9.6.7.1 Reserved—Bits 15–8

These bits are reserved or not implemented. They are read as, and written with 0s.

9.6.7.2 Source Address High (DMASAH)—Bits 7–0

This is the upper byte of the DMA Source Address. DMA data is read from this address. If post-incrementing (or decrementing) is enabled, this value is updated with each read transaction.

9.7 Programming Examples

Three different scenarios are described to illustrate the use of DMA.

9.7.1 Peripheral-to-Memory DMA Operation

In the first scenario, transfer data from an SPI peripheral to a 16-word block of data memory starting at address $1000 is desired. The SPI Data Receive Register is located at X:$1FFFEA. The SPI receive is assigned to DMA_REQ[1] and is provided in **Table 9-10.** In this scenario, assume the SPI is configured to transfer data by16-bit words. We would like the DMA to interrupt the core when the transfer is completed. To configure the DMA to perform this operation, follow these seven steps:

1. Write $001F to the DMA Source Address (high) register
2. Write $FFEA to the DMA Source Address (low) register
3. Write $0000 to the DMA Destination Address (high) register
4. Write $1000 to the DMA Destination Address (low) register
5. Write $0010 (hex = 16 decimal) to the DMA Transfer Count register
6. Write $0000 to the DMA Circular Queue Size register to disable the circular queue
7. Write $D012 to the DMA Transfer Control register

Steps 1 through 6 above may be performed in any order; however, step seven *must* be performed last. In step seven follow these steps:

- The DMA is enabled (DMAON = 1)
- DMA Interrupts are enabled (INTRON = 1)
- The DMA Interrupt Pending bit is cleared (INTRPEND = 0)
- The DMA data size is set to word (DATASIZE = 1)
- The SPI peripheral (DMA_REQ[1]) is selected for monitoring.
- The DMA source is set to not change values, to repeatedly read SPI data from the SPI Data Receive register (SSIM = 0)
- The DMA destination address is set to increment (CDSIM = 10) (by words because DATASIZE = 1)

9.7.2 Memory-to-Memory DMA Operation

In this scenario assume there are 256 bytes of data at address $1000 and the data is to be moved *in reversed order* to memory starting at $200 with a 256-byte buffer. **Figure 9-10** depicts this operation.

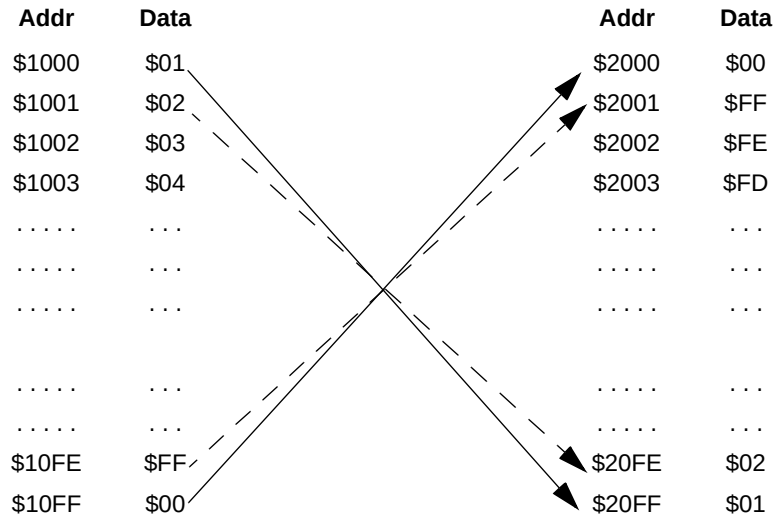


Figure 9-10. Memory-to-Memory DMA Mode

In this scenario, the DMA is to interrupt the core when the transfer is completed. To configure the DMA to perform this operation, follow these steps:

1. Write $0000 to the DMA Source Address (high) register
2. Write $1000 to the DMA Source Address (low) register (first write is from $00001000)
3. Write $0000 to the DMA Destination Address (high) register
4. Write $20FF to the DMA Destination Address (low) register (first write is from $000020FF)
5. Write $0100 to the DMA Transfer Count register (block is 256 bytes long)
6. Write $0000 to the DMA Circular Queue Size register (disable circular queue)
7. Write $C00B to the DMA Transfer Control register

Steps 1 through 6 above may be performed in any order; however, step seven *must* be performed last.

Note: Because this scenario wishes to store the byte sized data *in reverse order*, the initial destination address $2000 + $100 -1 = $20FF.

Note: The Transfer Control Register is written to increment the Source Address Register and *decrement* the Destination Address Register.

In step seven, the DMA is enabled (DAMON =1), DMA Interrupts are enabled (INTRON = 1), the DMA Interrupt Pending bit is cleared, the DMA data size is set to byte, $0 is written to the Peripheral Select field (indicating a memory-to-memory DMA operation), the DMA source is set to increment by bytes (CSSIM = 0), and the DMA destination address is set to *decrement* by bytes (DSIM = 11).

9.7.3 Peripheral-to-Memory Circular Queue DMA Operation

In this scenario, assume you want to transfer data from a SCI peripheral in a *continuous* transfer mode. The SCI0 data read register is at $1FFFE4. The SCI0 is assigned to DMA_REQ[3], illustrated in **Table 9-10.** The SCI is transferring one byte at a time, but it is stored as a word. You want the DMA to interrupt the core at the completion of each 40-word *packet*. The data is to be placed into a 400-word circular queue beginning at address $5000. **Figure 9-11** illustrates this operation.

Note: In the circular queue mode, the DMA operation runs endlessly until the DMA is explicitly turned off by the processor. If the byte mode is selected, source and destination addresses have no boundary restrictions.

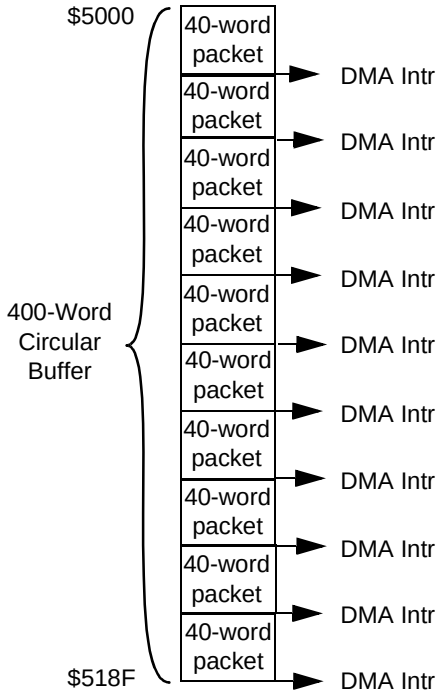


Figure 9-11. DMA Circular Queue Operation

To configure the DMA to perform this operation, the following steps would be taken:

1. Write $001F to the DMA Source Address (high) register
2. Write $FFE4 to the DMA Source Address (low) register
3. Write $0000 to the DMA Destination Address (high) register
4. Write $5000 to the DMA Destination Address (low) register
5. Write $0028 (hex = 40 decimal) to the DMA Transfer Count register
6. Write $0190 (hex = 400 decimal) to the DMA Circular Queue Size register
7. Write $D032 to the DMA Transfer Control register

Steps 1 through 6 above may be performed in any order; however, step seven *must* be performed last. In step seven follow these steps:

- The DMA is enabled (DMAON = 1)
- DMA Interrupts are enabled (INTRON = 1)
- The DMA Interrupt Pending bit is cleared (INTERPEND = 0)

- The DMA data size is set to word (DATASIZE = 1)
- The SCI0 peripheral (DMA_REQ[3]) is selected for monitoring.
- The DMA source is set to not change values to repeatedly read SCI data from the SCI0 Data Receive register (SSIM = 00)
- The DMA destination address is set to increment by words (DSIM = 10)

Chapter 10
Serial Communications Interface (SCI)

10.1 Introduction

This chapter describes the Serial Communications Interface (SCI) module. The module allows asynchronous serial communications with peripheral devices and other Digital Signal Controllers (DSCs).

10.2 Features

- Full-duplex or single wire operation
- Standard mark/space Non-Return-to-Zero (NRZ) format
- Thirteen-bit baud rate selection
- Programmable 8- or 9-bit data format
- Separately enabled transmitter and receiver
- Separate receiver and transmitter CPU interrupt requests
- Programmable polarity for transmitter and receiver
- Two receiver wake up methods:
  - idle line
  - address mark
- Interrupt-driven operation with seven flags:
  - transmitter empty
  - transmitter idle
  - receiver full
  - receiver overrun
  - noise error
  - framing error
  - parity error
- Receiver framing error detection
- Hardware parity checking
- 1/16 bit-time noise detection

10.3 Block Diagram

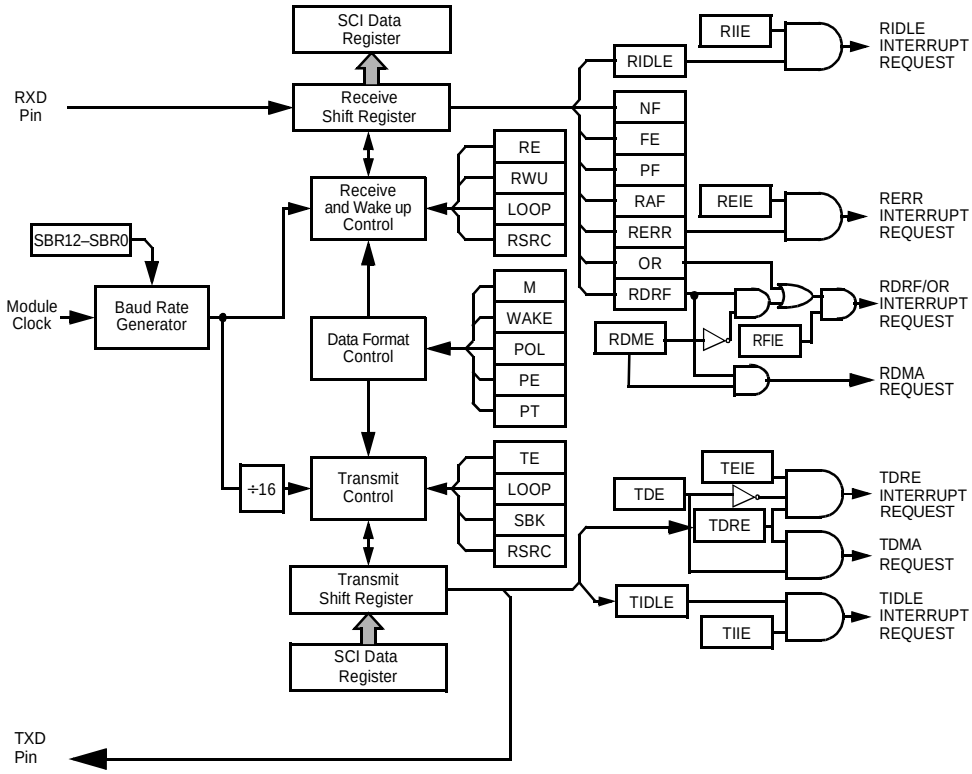


Figure 10-1. SCI Block Diagram

10.4 External Pin Descriptions

10.4.1 Transmit Data (TXD) Pin

The Transmit Data (TXD) pin is the SCI transmitter pin. TXD is available for general- purpose I/O when it is not configured for transmitter operation, such as TE = 0.

10.4.2 Receiver Data (RXD) Pin

The Receiver Data (RXD) pin is the SCI receiver pin. RXD is available for general- purpose I/O when it is not configured for receiver operation, such as RE = 0.

Data in **Table 10-1** are external I/O signals for the chip interface.

Table 10-1. External I/O Signals

Signal Name	I/O Type	Description	Reset State
TXD	Output	Transmit Data Pin	1
RXD	Input	Receiver Data Pin	—

10.5 Module Memory Maps

The five accessible registers on the SCI0 are listed in **Table 10-2** while the five accessible registers on the SCI1 are listed in **Table 10-3.**

Table 10-2. SCI0 Module Memory Map (SCI0_BASE = $1FFFE0)

Address Offset	Register Acronym	Register Name	Access Type	Chapter Location
Base + $0	SCI_0_BR	SCI Baud Rate Register	Read/Write	Section 10.9.1
Base + $1	SCI_0_CR	SCI Control Register	Read/Write	Section 10.9.2
Base + $2	SCI_0_CR2	SCI Control Register 2	Read/Write	Section 10.9.3
Base + $3	SCI_0_SR	SCI Status Register	*Read Only*	Section 10.9.4
Base + $4	SCI_0_DR	SCI Data Register	Read/Write	Section 10.9.5

Table 10-3. SCI1 Module Memory Map (SCI1_BASE = $1FFDF8)

Address Offset	Register Acronym	Register Name	Access Type	Chapter Location
Base + $0	SCI_1_BR	SCI Baud Rate Register	Read/Write	Section 10.9.1
Base + $1	SCI_1_CR	SCI Control Register	Read/Write	Section 10.9.2
Base + $2	SCI_1_CR2	SCI Control Register 2	Read/Write	Section 10.9.3
Base + $3	SCI_1_SR	SCI Status Register	*Read Only*	Section 10.9.4
Base + $4	SCI_1_DR	SCI Data Register	Read/Write	Section 10.9.5

Add. Offset	Register Name		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
$0	SCIBR	R	0	0	0	SBR												
		W																
$1	SCICR	R	LOOP	SWAI	RSRC	M	WAKE	POL	PE	PT	TEIE	TIIE	RFIE	REIE	TE	RE	RWU	SBK
		W																
$2	SCICR2	R	0	0	0	0	0	0	0	0	0	0	0	0	0	RIIE	TDE	RDE
		W																
$3	SCISR	R	TDRE	TIDLE	RDRF	RIDLE	OR	NF	FE	PF	0	0	0	0	0	0	0	RAF
		W																
$4	SCIDR	R	0	0	0	0	0	0	0	RECEIVE DATA								
		W								TRANSMIT DATA								

R 0 Read as 0
W Reserved

Figure 10-2. SCI Register Map Summary

10.6 Functional Description

Figure 10-1 explains the structure of the SCI module. The SCI allows full duplex, asynchronous, NRZ serial communication between the DSC and remote devices, including other digital signal controllers. The SCI transmitter and receiver operate independently, although they use the same baud rate generator. The monitors the status of the SCI, writes the data to be transmitted, and processes received data.

When initializing the SCI, be sure to set the proper peripheral enable bits in the GPIO registers as well as any pull-up enables.

10.6.1 Data Frame Format

The SCI uses the standard Non-Return-to-Zero (NRZ) mark/space data frame format illustrated in **Figure 10-3.**

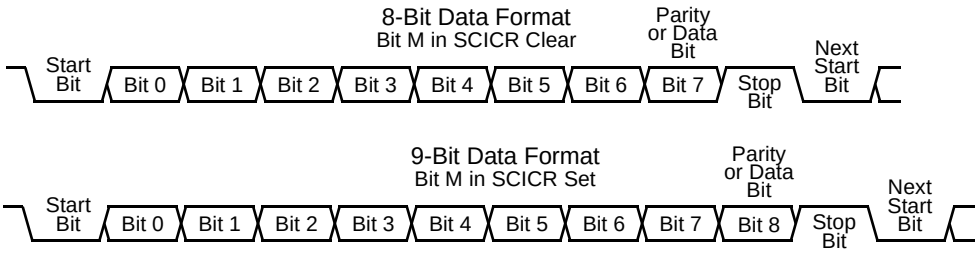


Figure 10-3. SCI Data Frame Formats

Each data character is contained in a frame including a Start bit, eight or nine Data bits, and a Stop bit. Clearing the M bit in the SCI Control Register (SCICR) configures the SCI for 8-bit data characters. A frame with eight data bits has a total of 10 bits.

Table 10-4. Example 8-Bit Data Frame Formats

Start Bit	Data Bits	Address Bit	Parity Bit	Stop Bit
1	8	0	0	1
1	7	0	1	1
1	7	1[1]	0	1

1. The address bit identifies the frame as an address character. Please see **Section 10.6.4.8**

Setting the M bit configures the SCI for 9-bit data characters. A frame with nine data bits has a total of 11 bits.

Table 10-5. Example 9-Bit Data Frame Formats

Start Bit	Data Bits	Address Bit	Parity Bit	Stop Bit
1	9	0	0	1
1	8	0	0	2
1	8	0	1	1
1	8	1[1]	0	1

1. The address bit identifies the frame as an address character. Please see **Section 10.6.4.8**

10.6.2 Baud Rate Generation

A 13-bit modulus counter in the baud rate generator derives the baud rate for both the receiver and the transmitter. A value of 1 to 8191 written to the SBR bits determines the module clock divisor. A value of zero disables the baud rate generator. The SBR bits are bits 12:0 of the SCI Baud Rate (SCIBR) register. The baud rate clock is synchronized with the bus clock, driving the receiver. The baud rate clock, divided by 16, drives the transmitter. The receiver has an acquisition rate of 16 samples per bit time.

Baud rate generation is subject to two sources of error:

1. Integer division of the module clock may not give the exact target frequency.
2. Synchronization with the bus clock can cause phase shift.

Table 10-6 lists examples of achieving target baud rates with a module clock frequency of 60MHz.

Table 10-6. Example Baud Rates (Module Clock = 60MHz)

SBR Bits	Receiver Clock (Hz)	Transmitter Clock (Hz)	Target Baud Rate	Error (%)
98	612245	38265	38,400	-0.35
195	307692	19231	19,200	0.16
391	153453	9591	9600	-0.09
781	76825	4802	4800	0.04
1563	38388	2399	2400	-0.04
3125	19200	1200	1200	0.00
6250	9600	600	600	0.00

Note: Maximum baud rate is module clock rate, divided by 16. System overhead may preclude processing the data at this speed.

10.6.3 SCI Transmitter Block Diagram

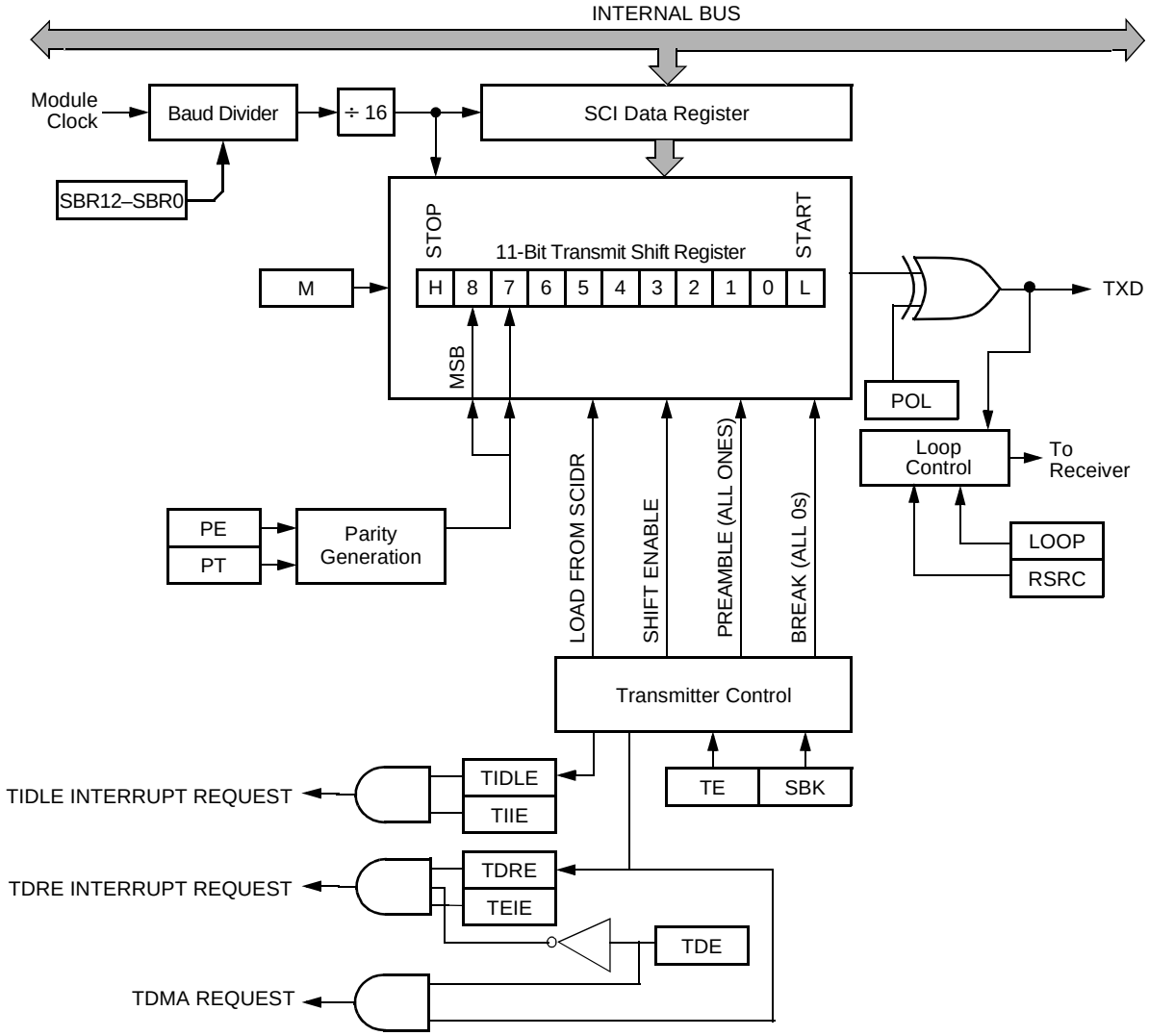


Figure 10-4. SCI Transmitter Block Diagram

10.6.3.1 Character Length

The SCI transmitter can accommodate either 8- or 9-bit data characters. The state of the M bit in the SCI Control Register (SCICR) determines the length of data characters.

10.6.3.2 Character Transmission

During an SCI transmission, the Transmit Shift Register shifts a frame out to the TXD pin. The data is written through the SCI data register.

To initiate an SCI transmission:

1. Enable the transmitter by writing a Logic 1 to the Transmitter Enable (TE) bit in the SCI Control Register (SCICR).

2. Clear the Transmit Data Register Empty (TDRE) flag by first reading the SCI Status Register (SCISR) and then writing output data to the SCI Data Register (SCIDR).
3. Repeat step 2 for each subsequent transmission.

Modifying the TE bit from 0 to a 1 automatically loads the Transmit Shift Register with a preamble of 10 logic 1s (if M = 0) or 11 logic 1s (if M = 1). After the preamble shifts out, control logic automatically transfers the data from the SCI Data Register into the Transmit Shift Register. A Logic 0 start bit automatically goes into the least significant bit position of the Transmit Shift Register. A Logic 1 Stop bit goes into the most significant bit position of the frame.

Hardware supports odd or even parity. When parity is enabled, the Most Significant Bit (MSB) of the data character is replaced by the parity bit.

The Transmit Data Register Empty (TDRE) flag in the SCI Status Register (SCISR) becomes set when the SCI Data Register transfers a character to the Transmit Shift Register. The TDRE flag indicates when the SCI Data Register can accept new data from the internal data bus. If the Transmitter Empty Interrupt Enable (TEIE) bit in the SCI Control Register (SCICR) is also set, the TDRE flag generates a transmitter interrupt request. If TDE is enabled, the DMA request will suppress the TDRE interrupt and a DMA request will be made instead.

When the Transmit Shift Register is not transmitting a frame and TE = 1, the TXD pin goes to the idle condition, Logic 1. If, at any time, software clears the TE bit in the SCI Control Register (SCICR), the transmitter relinquishes control of the port I/O pin upon completion of the current transmission causing the TXD pin to go to a HighZ state.

If software clears TE while a transmission is in progress (TIDLE = 0), the frame in the Transmit Shift Register continues to shift out. Then transmission stops even if there is data pending in the SCI Data Register. To avoid accidentally cutting off the last frame in a message, always wait for TDRE to go high after the last frame before clearing TE.

To separate messages with preambles with minimum idle line time, use this sequence between messages:

1. Write the last character of the first message to the SCIDR.
2. Wait for the TDRE flag to go high, indicating the transfer of the last frame to the Transmit Shift Register.
3. Queue a preamble by clearing and then setting the TE bit.
4. Write the first character of the second message to the SCIDR.

10.6.3.3 Break Characters

Writing a Logic 1 to the send break bit SBK in the SCI Control Register (SCICR) loads the Transmit Shift Register with a break character. A break character contains all logic 0s and has no start, stop, or parity bit. Break character length depends on the M bit in the SCI Control Register (SCICR). As long as SBK is at Logic 1, transmitter logic continuously loads break characters into the Transmit Shift register. After software clears the SBK bit, the Shift register finishes transmitting the last break character and then transmits at least one Logic 1. The automatic Logic 1 at the end of the last break character guarantees the recognition of the Start bit of the next frame.

The SCI recognizes a break character when a start bit is followed by eight or nine Logic 0 data bits and a Logic 0 where the Stop bit should be. Receiving a break character has these effects on SCI registers:

- Sets the Framing Error (FE) flag
- Sets the Receive Data Register Full (RDRF) flag
- Clears the SCI Data Register (SCIDR)
- May set the Overrun (OR) flag, Noise Flag (NF), Parity Error (PE) flag, or the Receiver Active Flag (RAF). Please see the SCI Status Register in **Section 10.9.4.**

10.6.3.4 Preambles

A preamble contains all logic 1s and has no start, stop, or parity bit. A preamble length depends on the M bit in the SCI Control Register (SCICR). The preamble is a synchronizing mechanism initiating the first transmission begun after modifying the TE bit from 0 to 1.

If the TE bit is cleared during a transmission, the TXD pin becomes idle after completion of the transmission in progress. Clearing and then setting the TE bit during a transmission queues a preamble to be sent after the frame currently being transmitted.

Note: Toggle the TE bit for a queued preamble when the TDRE flag becomes set and immediately before writing the next character to the SCI Data Register.

When queueing a preamble, return the TE bit to Logic 1 before the stop bit of the current frame shifts out to the TXD pin. Setting TE after the stop bit appears on TXD causes data previously written to the SCI Data Register to be lost.

10.6.3.5 Receiver

Figure 10-5 explains the block diagram of the SCI receiver with detailed discussion of the receiver function in the following paragraphs.

10.6.4 SCI Receiver Block Diagram

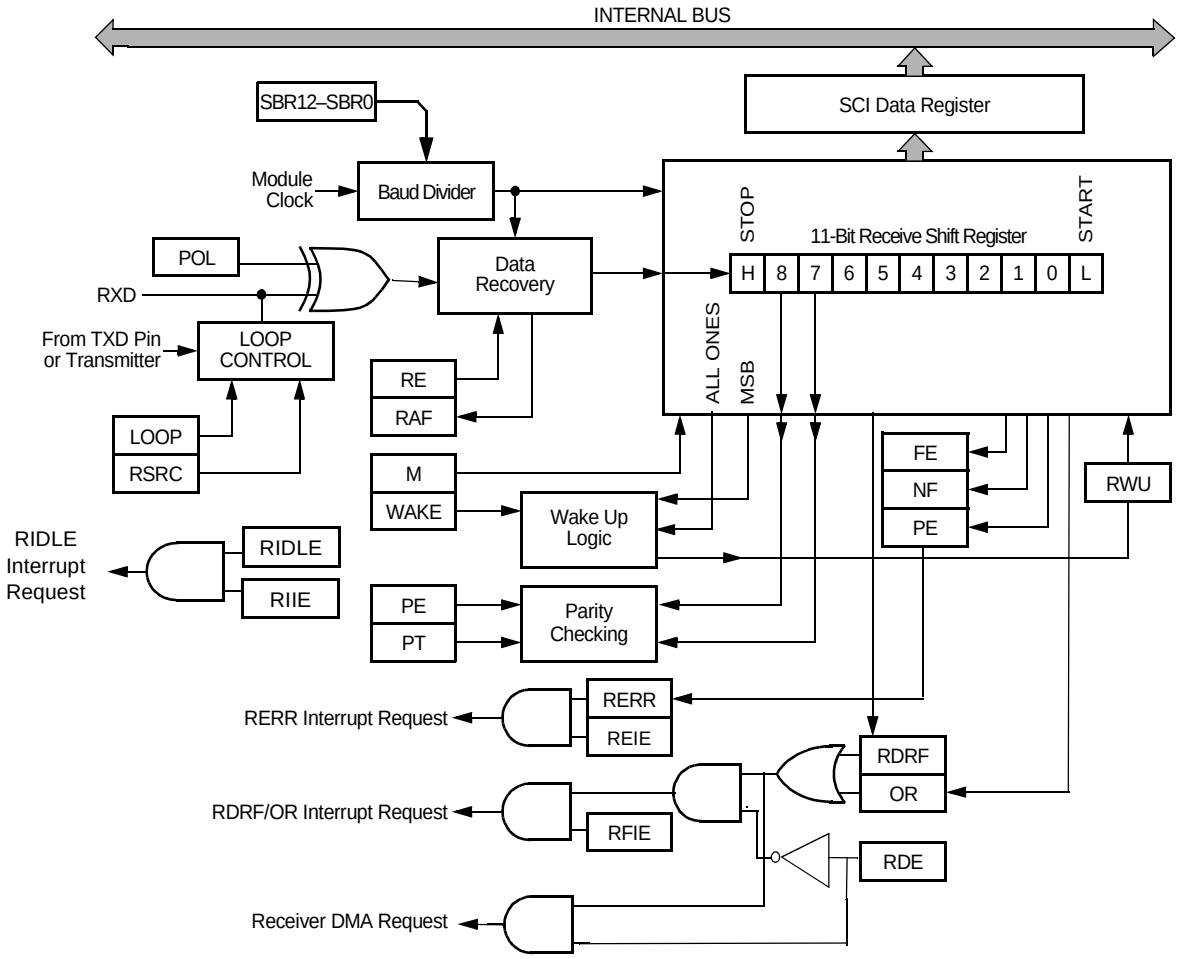


Figure 10-5. SCI Receiver Block Diagram

10.6.4.1 Character Length

The SCI receiver can accommodate either 8- or 9-bit data characters. The state of the M bit in the SCI Control Register (SCICR) determines the length of data characters.

10.6.4.2 Character Reception

During an SCI reception, the Receive Shift register shifts a frame in from the RXD pin. The data is read from the SCI Data Register (SCIDR).

After a complete frame shifts into the Receive Shift register, the data portion of the frame transfers to the SCI Data Register. The Receive Data Register Full (RDRF) flag in the SCI Status Register (SCISR) becomes set, indicating the received character can be read. If the Receive Full Interrupt Enable (RFIE) bit in the SCI Control Register (SCICR) is also set, the RDRF flag generates an RDRF interrupt request. If RDE is set, the RDRF interrupt is suppressed and a Receive DMA request is generated instead.

10.6.4.3 Data Sampling

The receiver samples the RXD pin at the RT clock rate. The RT clock is an internal signal with a frequency 16 times the baud rate. To adjust for baud rate mismatch, the RT clock illustrated in **Figure 10-6** is resynchronized:

- After every start bit
- After the receiver detects a data bit change from Logic 1 to Logic 0 (after the majority of data bit samples at RT8, RT9, and RT10 returns a valid Logic 1 and the majority of the next RT8, RT9, and RT10 samples returns a valid Logic 0)

To locate the Start bit, data recovery logic does an asynchronous search for a Logic 0 preceded by three logic 1s. When the falling edge of a possible Start bit occurs, the RT clock begins to count to 16.

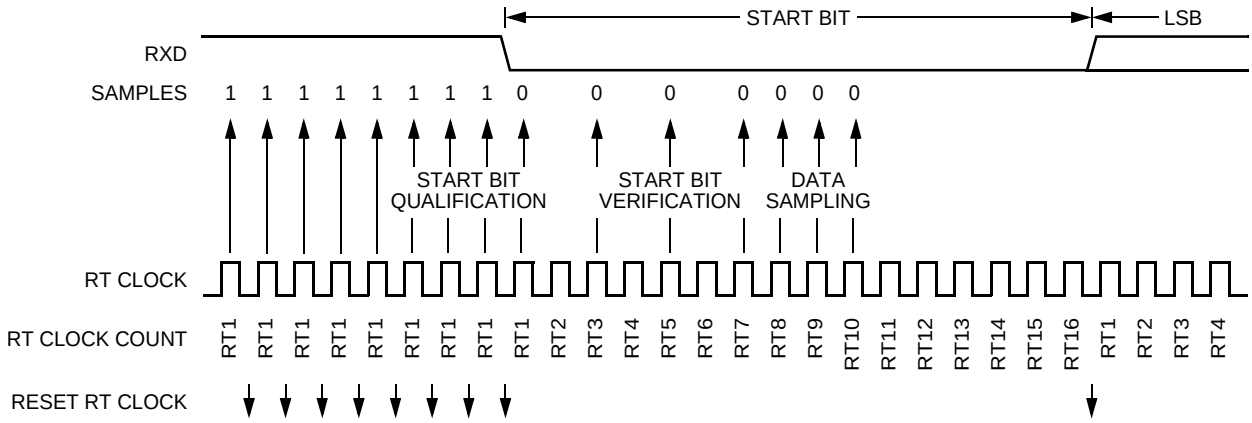


Figure 10-6. Receiver Data Sampling

To verify the Start bit and to detect noise, data recovery logic takes samples at RT3, RT5, and RT7. **Table 10-7** summarizes the results of the Start bit verification samples. If the Start bit verification is not successful, the RT clock is reset and a new search for a Start bit begins.

Table 10-7. Start Bit Verification

RT8, RT9, and RT10 Samples	Data Bit Determination	Noise Flag
000	Yes	0
001	Yes	1
010	Yes	1
011	No	0
100	Yes	1
101	No	0
110	No	0
111	No	0

If Start bit verification is not successful, the RT clock is reset and a new search for a Start bit begins.

To determine the value of a Data bit and to detect noise, recovery logic takes samples at RT8, RT9, and RT10. **Table 10-8** summarizes the results of the Data bit samples.

Table 10-8. Data Bit Recovery

RT8, RT9, and RT10 Samples	Data Bit Determination	Noise Flag
000	0	0
001	0	1
010	0	1
011	1	1
100	0	1
101	1	1
110	1	1
111	1	0

Note: The RT8, RT9, and RT10 samples do not affect Start bit verification. If any or all of the RT8, RT9, and RT10 Start bit samples are logic 1s following a successful Start bit verification, the Noise Flag (NF) is set and the receiver assumes that the bit is a Start bit (Logic 0).

To verify a Stop bit and to detect noise, recovery logic takes samples at RT8, RT9, and RT10. **Table 10-9** summarizes the results of the Stop bit samples.

Table 10-9. Stop Bit Recovery

RT8, RT9, and RT10 Samples	Framing Error Flag	Noise Flag
000	1	0
001	1	1
010	1	1
011	0	1
100	1	1
101	0	1
110	0	1
111	0	0

Figure 10-7 illustrates the verification samples RT3 and RT5 determine the first low detected was noise and not the beginning of a Start bit. The RT clock is reset and the Start bit search begins again. The Noise Flag is not set because the noise occurred before the Start bit was found.

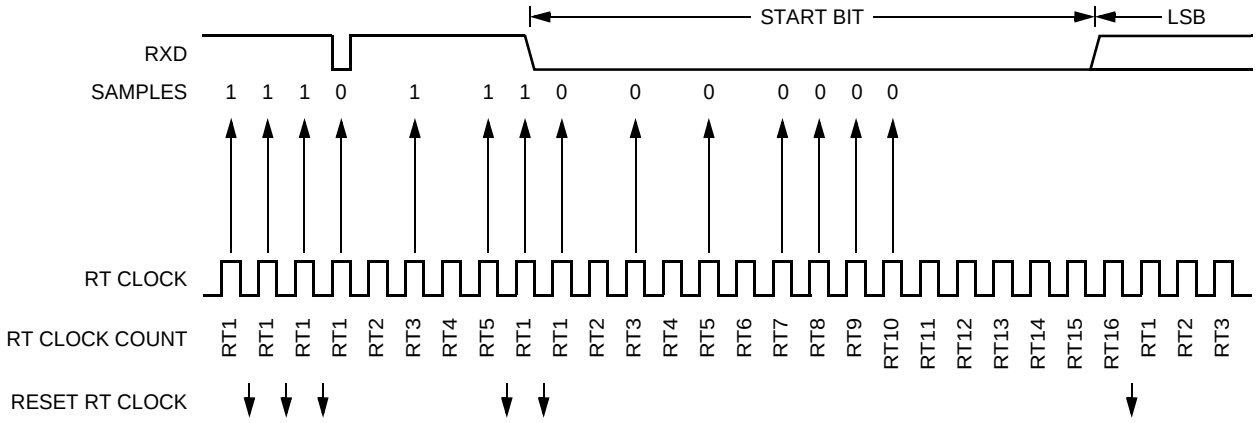


Figure 10-7. Start Bit Search Example 1

Figure 10-8 shows noise is perceived as the beginning of a Start bit although the verification sample at RT3 is high. The RT3 sample sets the Noise Flag. Although the perceived bit time is misaligned, the data samples RT8, RT9, and RT10 are within the bit time and data recovery is successful.

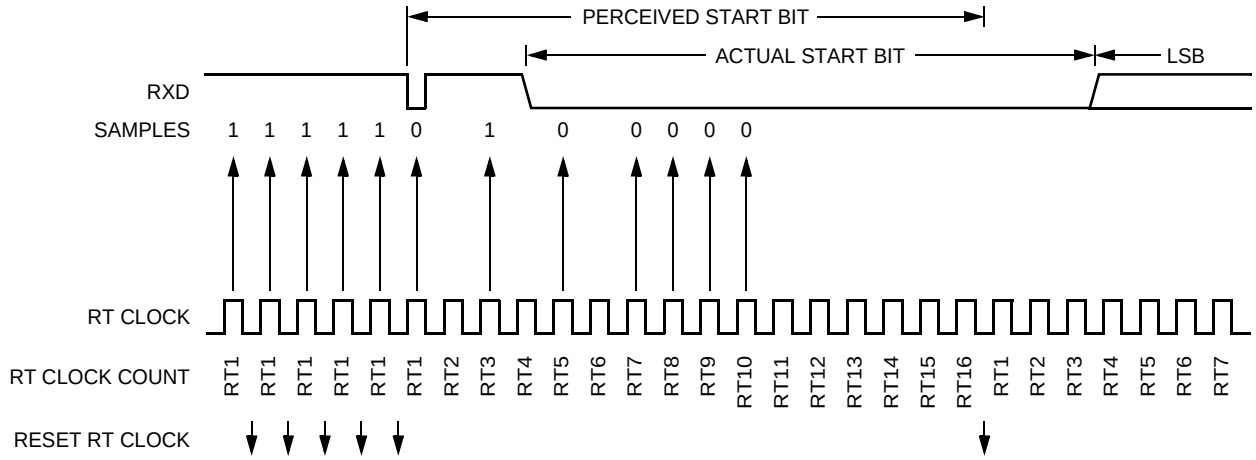


Figure 10-8. Start Bit Search Example 2

Figure 10-9 illustrates a large burst of noise is perceived as the beginning of a Start bit, although the test sample at RT5 is high. The RT5 sample sets the Noise Flag. Although this is a worst-case misalignment of perceived bit time, the data samples RT8, RT9, and RT10 are within the bit time and data recovery is successful.

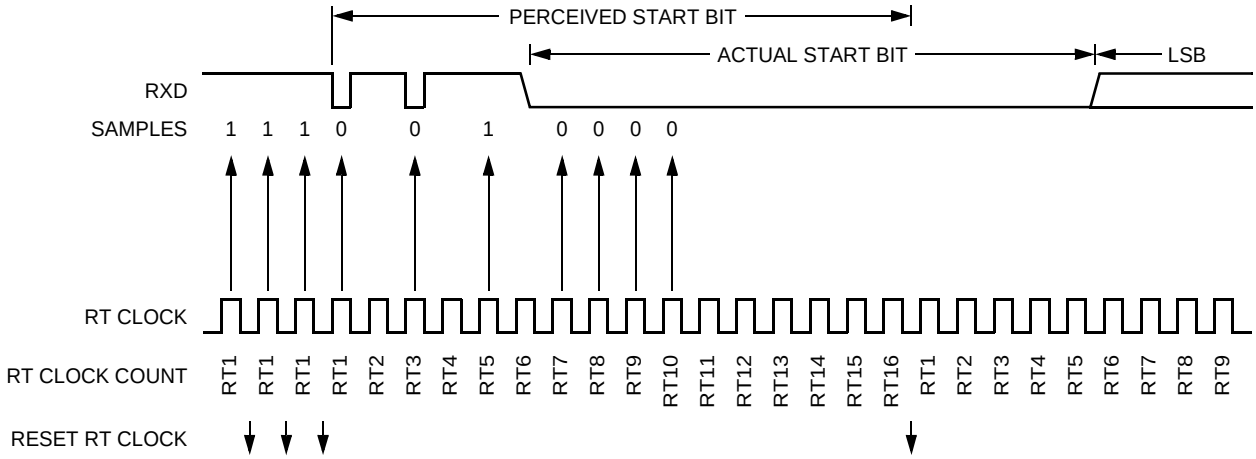


Figure 10-9. Start Bit Search Example 3

Figure 10-10 illustrates the effect of noise early in the Start bit time. Although this noise does not affect proper synchronization with the Start bit time, it does set the Noise Flag.

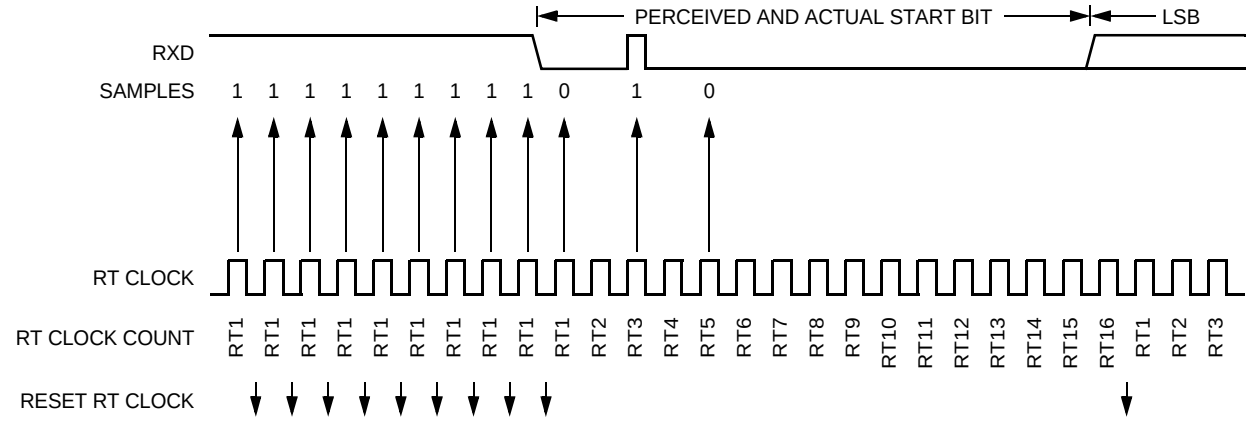


Figure 10-10. Start Bit Search Example 4

Figure 10-11 demonstrates a burst of noise near the beginning of the Start bit that resets the RT clock. The sample after the reset is low but is not preceded by three high samples that would qualify as a falling edge. Depending on the timing of the Start bit search and on the data, the frame may be missed entirely or it may set the framing error flag.

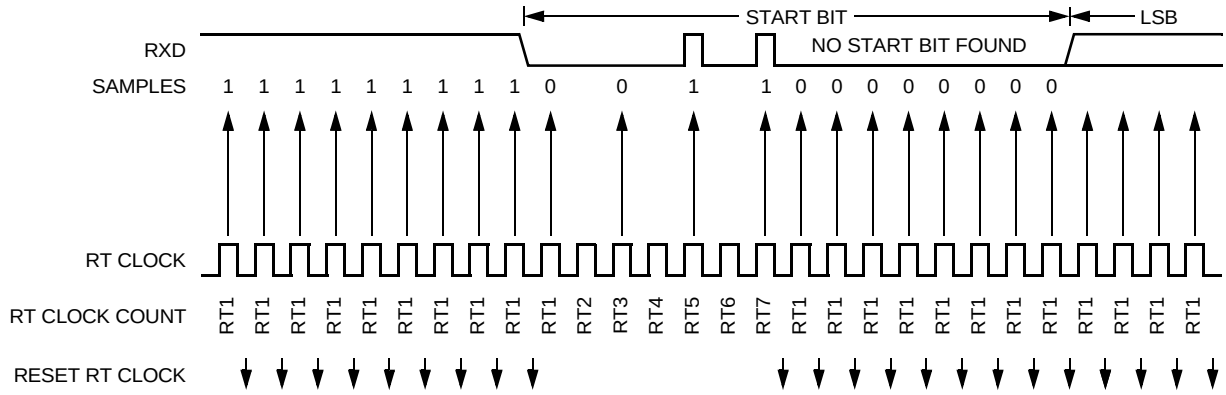


Figure 10-11. Start Bit Search Example 5

Figure 10-12 shows a noise burst makes the majority of data samples RT8, RT9, and RT10 high. This sets the Noise Flag but does not reset the RT clock. In Start bits only, the RT8, RT9, and RT10 data samples are ignored.

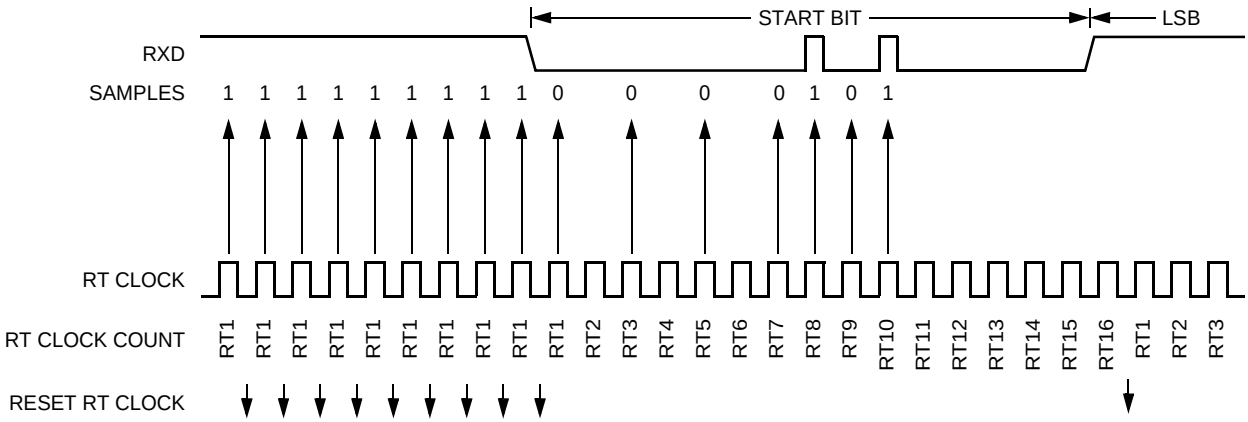


Figure 10-12. Start Bit Search Example 6

10.6.4.4 Framing Errors

If the data recovery logic does not detect a Logic 1 where the Stop bit should be in an incoming frame, it sets the Framing Error (FE) flag in the SCI Status Register (SCISR). A break character also sets the FE flag because a break character has no Stop bit. The FE flag is set at the same time as the RDRF flag. The FE flag inhibits further data reception until it is cleared.

10.6.4.5 Baud Rate Tolerance

A transmitting device may be operating at a baud rate below or above the receiver baud rate. Accumulated bit time misalignment can cause one of the three Stop bit data samples to fall outside the actual Stop bit. Then a noise error occurs. If more than one of the samples is outside the Stop bit, a framing error occurs. In most applications, the baud rate tolerance is much more than the degree of misalignment that is likely to occur.

As the receiver samples an incoming frame, it resynchronizes the RT clock on any valid falling edge within the frame. Re-synchronization within frames corrects misalignments between transmitter bit times and receiver bit times.

10.6.4.6 Slow Data Tolerance

Figure 10-13 explains how much a slow received frame can be misaligned without causing a noise error or a framing error. The slow Stop bit begins at RT8 instead of RT1, it arrives in time for the Stop bit data samples at RT8, RT9, and RT10.

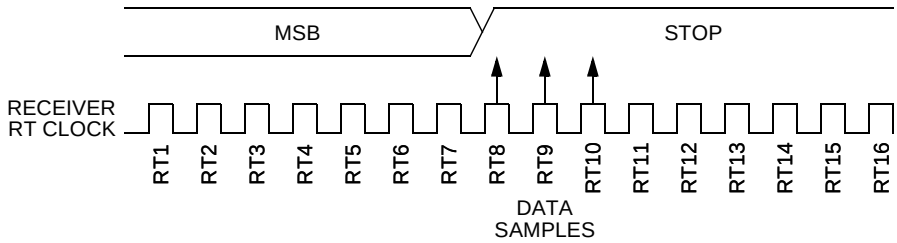


Figure 10-13. Slow Data

For an 8-bit data character, data sampling of the Stop bit takes the receiver 9-bit × 16 RT cycles + 10 RT cycles = 154 RT cycles.

With the misaligned character shown in **Figure 10-13**, the receiver counts 154 RT cycles at the point when the count of the transmitting device is 9-bit × 16 RT cycles + 3 RT cycles = 147 RT cycles.

The maximum percent difference between the receiver count and the transmitter count of a slow 8-bit data character with no errors is:

$$\left|\frac{154 - 147}{154}\right| \times 100 = 4.54\%$$

For a 9-bit data character, data sampling of the Stop bit takes the receiver 10-bit × 16 RT cycles + 10 RT cycles = 170 RT cycles.

With the misaligned character, shown in **Figure 10-13,** the receiver counts 170 RT cycles at the point when the count of the transmitting device is10 bit × 16 RT cycles + 3 RT cycles = 163 RT cycles.

The maximum percent difference between the receiver count and the transmitter count of a slow 9-bit character with no errors is:

$$\left|\frac{170 - 163}{170}\right| \times 100 = 4.12\%$$

10.6.4.7 Fast Data Tolerance

Figure 10-14 demonstrates how much a fast received frame can be misaligned without causing a noise error or a framing error. The fast Stop bit ends at RT10 instead of RT16 but it is still sampled at RT8, RT9, and RT10.

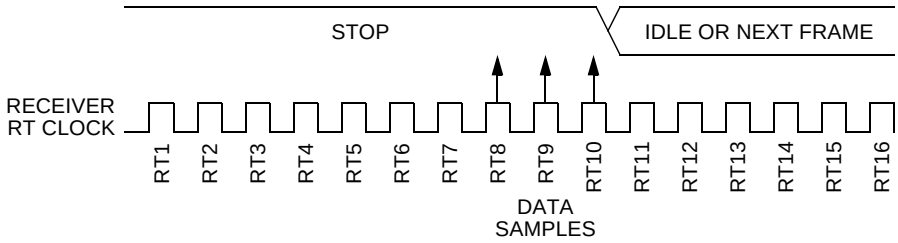


Figure 10-14. Fast Data

For an 8-bit data character, data sampling of the Stop bit takes the receiver 9-bit × 16 RT cycles + 10 RT cycles = 154 RT cycles.

With the misaligned character shown in **Figure 10-14,** the receiver counts 154 RT cycles at the point when the count of the transmitting device is 10 bit × 16 RT cycles = 160 RT cycles.

The maximum percent difference between the receiver count and the transmitter count of a fast 8-bit character with no errors is:

$$\left|\frac{154 - 160}{154}\right| \times 100 = 3.90\%$$

For a 9-bit data character, data sampling of the stop bit takes the receiver 10 bit × 16 RT cycles + 10 RT cycles = 170 RT cycles.

With the misaligned character shown in **Figure 10-14**, the receiver counts 170 RT cycles at the point when the count of the transmitting device is 11 bit × 16 RT cycles = 176 RT cycles.

The maximum percent difference between the receiver count and the transmitter count of a fast 9-bit character with no errors is:

$$\left|\frac{170 - 176}{170}\right| \times 100 = 3.53\%$$

10.6.4.8 Receiver Wake Up

In order for the SCI to ignore transmissions intended only for other receivers in multiple- receiver systems, the receiver can be placed into a standby state. Setting the Receiver Wake Up (RWU) bit in the SCI Control Register (SCICR) places the receiver into a standby state while receiver interrupts are disabled.

The transmitting device can address messages to selected receivers by including addressing information in the initial frame or frames of each message.

The WAKE bit in the SCI Control Register (SCICR) determines how the SCI is brought out of the standby state to process an incoming message. The WAKE bit enables either idle line wake up or address mark wake up:

- Idle Input Line Wake Up (WAKE = 0)—In this wake up method, an idle condition on the RXD pin clears the RWU bit and wakes up the SCI. The initial frame or frames of every message contain addressing information. All receivers evaluate the addressing information, and receivers for which the message is addressed process the following frames. Any receiver for which a message is not addressed can set its RWU bit and return to the standby state. The RWU bit remains set and the receiver remains on standby until another preamble appears on the RXD pin.

  Idle line wake up requires messages be separated by at least one preamble and no message contains preambles.

  The preamble waking a receiver does not set the receiver Idle (IDLE) bit or the Receive Data Register Full (RDRF) flag.

- Address Mark Wake up (WAKE = 1)—In this wake up method, a Logic 1 in the Most Significant Bit (MSB) position of a frame clears the RWU bit and wakes up the SCI. The Logic 1 in the MSB position marks a frame as an address frame, containing the addressing information. All receivers evaluate the addressing information as well as the receivers for which the message is addressed in the following frames. Any receiver for which a message is not addressed can set its RWU bit and return to the standby state. The RWU bit remains set and the receiver remains on standby until another address frame appears on the RXD pin.

The Logic 1 MSB of an address frame clears the receiver’s RWU bit before the Stop bit is received, setting the RDRF flag.

Address Mark Wake Up allows messages to contain preambles but it requires the MSB to be reserved for use in address frames.

Note: With the WAKE bit clear, setting the RWU bit after the RXD pin has been idle can cause the receiver to wake up immediately.

10.6.5 Single Wire Operation

Normally, the SCI uses two pins for transmitting and receiving. In the single wire operation, the RXD pin is disconnected from the SCI and is available as a General Purpose I/O (GPIO) pin. The SCI uses the TXD pin for both receiving and transmitting.

Setting the TE bit in the SCI Control Register (SCICR) configures TXD as the output for transmitted data. Clearing the TE bit configures TXD as the input for received data.

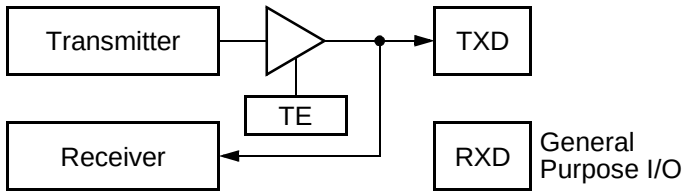


Figure 10-15. Single Wire Operation (LOOP = 1, RSRC = 1)

Enable single wire operation by setting the LOOP bit and the Receiver Source (RSRC) bit in the SCI Control Register (SCICR). Setting the LOOP bit disables the path from the RXD pin to the receiver. Setting the RSRC bit connects the receiver input to the output of the TXD pin driver.

10.6.6 Loop Operation

In Loop operation the transmitter output goes to the receiver input. The RXD pin is disconnected from the SCI and is available as a GPIO pin.

Setting the TE bit in the SCI Control Register (SCICR) connects the transmitter output to the TXD pin. Clearing the TE bit disconnects the transmitter output from the TXD pin.

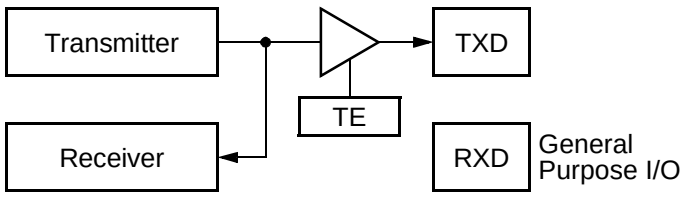


Figure 10-16. Loop Operation (LOOP = 1, RSRC = 0)

Enable Loop operation by setting the LOOP bit and clearing the RSRC bit in the SCI Control Register (SCICR). Setting the LOOP bit disables the path from the RXD pin to the receiver. Clearing the RSRC bit connects the transmitter output to the receiver input. Both the transmitter and receiver must be enabled (TE = 1 and RE = 1).

10.7 DMA Operation

10.7.1 Transmit DMA Operation

Setting the TDE bit in SCICR2 enables Transmit DMA mode. In this mode, the Transmitter Empty Interrupt is suppressed, allowing a transmitter DMA request to be generated. The DMA controller can be configured to write to the SCIDR, clearing the TDRE and the DMA request.

10.7.2 Receive DMA Operation

Setting the RDE bit in SCICR2 enables Receiver DMA mode. In this mode, the Receiver Full Interrupt is suppressed, allowing a Receiver DMA request to be generated. The DMA controller can then be configured to read to the SCIDR, clearing the interrupt.

10.7.3 Receiver Wake Up with DMA

If DMA operation is desired during either of the wake up modes of operation, DMA requests should only be made for a message addressed to the receiver. For example, Wake Up operation proceeds normally until a desired receive message is identified. Then DMA requests are generated to buffer the remainder of the message. An example of the DMA receive operations is provided in **Appendix 10-10, “Receiver Wake Up with DMA.” .**

Table 10-10. Receiver Wake Up with DMA

1. Configure the SCI for standard receive operation 2. SCI receives the first frame of the incoming message and stores it in the SCIDR 3. A SCI Receiver Full interrupt occurs 4. The ISR looks at the message address information and determines:	
MESSAGE NOT FOR US	MESSAGE IS FOR US
5. Configure SCI for RWU mode and wait for the end of the message. Repeat from setup 1.	5. Enable DMA operation with a buffer large enough to accommodate the max message size. 6. Enable RIIE so the SCI will interrupt at the completion of the message. (Assumes the DMA buffer does not fill up first.) 7. When the RII interrupt occurs, process the message and return to setup 1.

10.8 Low Power Modes

10.8.1 Run Mode

Clearing the Transmitter Enable (TE) or Receiver Enable (RE) bits in the SCI Control Register (SCICR) reduces power consumption in the Run mode. SCI registers are still accessible when TE or RE is cleared, but clocks to the core of the SCI are disabled.

10.8.2 Wait Mode

SCI operation in the Wait mode depends on the state of the SWAI bit in the SCI Control Register (SCICR).

- If the SWAI bit is clear, the SCI operates normally when the Central Processing Unit (CPU) is in the Wait mode.
- If the SWAI bit is set, SCI clock generation ceases and the SCI module enters a power-conservation state when the CPU is in the Wait mode. In this condition, SCI registers are not accessible. Setting SWAI does not affect the state of the Receiver Enable (RE) bit or the Transmitter Enable (TE) bit.

  If the SWAI bit is set, any transmission or reception in progress stops at the Wait mode entry. The transmission or reception resumes when either an internal or external interrupt brings the digital signal controller out of the Wait mode. Exiting the Wait mode by reset aborts any transmission or reception in progress and resets the SCI.

10.8.3 Stop Mode

The SCI is inactive in the Stop mode for reduced power consumption. The STOP instruction does not affect SCI register states. SCI operation resumes after an external interrupt brings the CPU out of the Stop mode. Exiting the Stop mode by reset aborts any transmission or reception in progress and resets the SCI.

10.8.4 Wait Mode Recovery

Any enabled SCI interrupt request can bring the CPU out of the Wait mode.

10.9 SCI Register Descriptions (SCI0_BASE = $1FFFE0 and SCI1_BASE = $1FFDF8)

10.9.1 SCI Baud Rate (SCIBR)

This register can be read at anytime. Bits 12 through zero can be written at any time, but bits 15 through 13 are reserved and can be modified only in special modes.

Base + $0	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	0	0	0	SBR												
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 10-17. SCI Baud Rate Register (SCIBR)

See Programmer's Sheet on Appendix page B-53

The count in this register determines the baud rate of the SCI. The formula for calculating baud rate is:

$$\text{SCI baud rate} = \frac{\text{SCI module clock}}{16 \times \text{SBR}}$$

SBR = contents of the baud rate registers, a value of 1 to 8191

Note: The baud rate generator is disabled until the TE or the RE bits are set for the first time after reset. The baud rate generator is disabled when SBR = 0.

10.9.2 SCI Control Register (SCICR)

The SCI Control Register can be read/written at anytime.

Base + $1	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	LOOP	SWAI	RSRC	M	WAKE	POL	PE	PT	TEIE	TIIE	RFIE	REIE	TE	RE	RWU	SBK
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 10-18. SCI Control Register (SCICR)

See Programmer's Sheets on Appendix page B-54

10.9.2.1 Loop Select Bit (LOOP)—Bit 15

This bit enables Loop operation. Loop operation disconnects the RXD pin from the SCI and the transmitter output goes into the receiver input. Both the transmitter and receiver must be enabled to use the internal Loop function as opposed to single wire operation, requiring only one or the other to be enabled. Please see **Table 10-11.**

- 0 = Normal operation enabled
- 1 = Loop operation enabled

The receiver input is determined by the RSRC bit. The transmitter output is controlled by the TE bit.

If the TE bit is set and LOOP = 1, the transmitter output appears on the TXD pin. If the TE bit is clear and LOOP = 1, the TXD pin is high-impedance.

Table 10-11. Loop Functions

LOOP	RSRC	Function
0	X	Normal operation
1	0	Loop mode with internal TXD fed back to RXD
1	1	Single-wire mode with TXD output fed back to RXD

10.9.2.2 Stop in Wait Mode (SWAI)—Bit 14

This bit disables the SCI in the Wait mode.

- 0 = SCI enabled in Wait mode
- 1 = SCI disabled in Wait mode

10.9.2.3 Receiver Source (RSRC)— Bit 13

When LOOP = 1, the RSRC bit determines the internal feedback path for the receiver.

- 0 = Receiver input internally connected to transmitter output
- 1 = Receiver input connected to TXD pin

10.9.2.4 Data Format Mode (M)—Bit 12

This bit determines whether data characters are eight or nine bits long.

- 0 = One Start bit, eight data bits, one Stop bit
- 1 = One Start bit, nine data bits, one Stop bit

10.9.2.5 Wake Up Condition (WAKE)—Bit 11

This bit determines which condition wakes up the SCI: a Logic 1 (address mark) in the MSB position of a received data character or an idle condition on the RXD pin.

- 0 = Idle line wake up
- 1 = Address mark wake up

10.9.2.6 Polarity (POL)—Bit 10

This bit determines whether to invert the data as it goes from the transmitter to the TXD pin and from the RXD pin to the receiver. All bits, Start, Data, and Stop, will be inverted as they leave the Transmit Shift Register and before they enter the Receive Shift register.

- 0 = Doesn’t invert Transmit and Receive Data bits (Normal mode)
- 1 = Invert Transmit and Receive Data bits (Inverted mode)

Note: It is recommended the POL bit be toggled only when both TE and RE = 0.

10.9.2.7 Parity Enable (PE)—Bit 9

This bit enables the parity function. When enabled, the parity function replaces the MSB of the data character with a parity bit.

- 0 = Parity function disabled
- 1 = Parity function enabled

10.9.2.8 Parity Type (PT)—Bit 8

This bit determines whether the SCI generates and checks for even parity or odd parity of the data bits. With even parity, an *even* number of *ones*, clears the parity bit. An *odd* number of *ones* sets the parity bit. With odd parity, an *odd* number of *ones*, clears the parity bit and an *even* number of *ones* sets the parity bit.

- 0 = Even parity
- 1 = Odd parity

10.9.2.9 Transmitter Empty Interrupt Enable (TEIE)—Bit 7

This bit enables the Transmit Data Register Empty (TDRE) flag to generate interrupt requests.

- 0 = TDRE interrupt requests disabled
- 1 = TDRE interrupt requests enabled

10.9.2.10 Transmitter Idle Interrupt Enable (TIIE)—Bit 6

This bit enables the Transmitter Idle (TIDLE) flag to generate interrupt requests.

- 0 = TIDLE interrupt requests disabled
- 1 = TIDLE interrupt requests enabled

10.9.2.11 Receiver Full Interrupt Enable (RFIE)—Bit 5

This bit enables the Receive Data Register Full (RDRF) flag, or the Overrun (OR) flag to generate interrupt requests.

- 0 = RDRF and OR interrupt requests disabled
- 1 = RDRF and OR interrupt requests enabled

10.9.2.12 Receive Error Interrupt Enable (REIE)—Bit 4

This bit enables the Receive Error (RE) flags (NF, PF, FE, and OR) to generate interrupt requests.

- 0 = Error interrupt requests disabled
- 1 = Error interrupt requests enabled

10.9.2.13 Transmitter Enable (TE)—Bit 3

This bit enables the SCI transmitter and configures the TXD pin as the SCI transmitter output. The TE bit can be used to queue an idle preamble.

- 0 = Transmitter disabled
- 1 = Transmitter enabled

10.9.2.14 Receiver Enable (RE)—Bit 2

This bit enables the SCI Receiver.

- 0 = Receiver disabled
- 1 = Receiver enabled

10.9.2.15 Receiver Wake Up (RWU)—Bit 1

This bit enables the wake up function, inhibiting further receiver interrupt requests. Normally, hardware wakes the receiver by automatically clearing the RWU. Please refer to **Section 10.6.4.8** for a description of Receive Wake Up.

- 0 = Normal operation
- 1 = Standby state

10.9.2.16 Send Break (SBK)—Bit 0

Toggling SBK sends one break character (10 or 11 logic 0s). As long as SBK is set, the transmitter sends logic 0s.

- 0 = No break characters
- 1 = Transmit break characters

10.9.3 SCI Control Register 2 (SCICR2)

This register can be read/written at anytime.

Base + $2	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	0	0	0	0	0	0	0	0	0	0	0	0	0	RIIE	TDE	RDE
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 10-19. SCI Control Register 2 (SCICR2)

See Programmer's Sheets on Appendix page B-57

10.9.3.1 Reserved—Bits 15–3

These bits are reserved or not implemented. They are read as 0, but cannot be modified by writing.

10.9.3.2 Receiver Idle Interrupt Enable (RIIE)—Bit 2

This read/write RIIE bit is used to let the processor know a message has completed when using DMA in a wake up mode of operation. The receiver is configured normally until a message is detected. The processor code then determines if the message is for it. If so, DMA is enabled for the maximum message size, and the RIIE interrupt is enabled to tell the core when the message has completed. The receive DMA operation would be halted by the interrupt service routine at this time.

10.9.3.3 Transmitter DMA Enable (TDE)—Bit 1

TDE enables DMA mode transfer of data from the SCIDR. For transmit DMA operation, this bit must be set and a DMA channel must be configured to utilize the request.

- 0 = Transmit DMA disabled
- 1 = Transmit DMA enabled

10.9.3.4 Receiver DMA Enable (RDE)—Bit 0

RDE enables DMA mode transfer of data to the SCIDR. For receive DMA operation, this bit must be set and a DMA channel must be configured to utilized the request.

- 0 = Receive DMA disabled
- 1 = Receive DMA enabled

10.9.4 SCI Status Register (SCISR)

This register can be read at anytime; however, it cannot be modified by writing. Writes clear flags.

Base + $3	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	TDRE	TIDLE	RDRF	RIDLE	OR	NF	FE	PF	0	0	0	0	0	0	0	RAF
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 10-20. SCI Status Register (SCISR)

See Programmer's Sheets on Appendix page B-58

10.9.4.1 Transmit Data Register Empty Flag (TDRE)—Bit 15

This bit is set when the Transmit Shift register receives a character from the SCI Data Register (SCIDR). Clear TDRE by reading SCISR with TDRE set and then writing to the SCI Data register in Normal mode or by writing the SCIDR with TDE set.

- 0 = No character transferred to Transmit Shift register
- 1 = Character transferred to Transmit Shift register; Transmit Data register empty

10.9.4.2 Transmitter Idle Flag (TIDLE)—Bit 14

This bit is set when the TDRE flag is set and no data, preamble, or break character is being transmitted. When TIDLE is set, the TXD pin becomes idle (Logic 1). Clear TIDLE by reading the SCI Status Register (SCISR) with TIDLE set and then writing to the SCI Data Register (SCIDR). TIDLE is not generated when a data character, a preamble, or a break is queued and ready to be sent.

- 0 = Transmission in progress
- 1 = No transmission in progress

10.9.4.3 Receive Data Register Full Flag (RDRF)—Bit 13

This bit is set when the data in the Receive Shift register transfers to the SCI Data register (SCIDR). Clear RDRF by reading the SCI Status Register (SCISR) with RDRF set and then reading the SCI Data register in Normal mode or by reading the SCIDR with RDE set.

- 0 = Data not available in SCI Data register
- 1 = Received data available in SCI Data register

10.9.4.4 Receiver Idle Line Flag (RIDLE)—Bit 12

This bit is set when 10 consecutive logic 1s (if M = 0) or 11 consecutive logic 1s (if M = 1) appear on the receiver input. Once the RIDLE flag is cleared (the receiver detects a Logic 0), a valid frame must again set the RDRF flag before an idle condition can set the RIDLE flag.

- 0 = Receiver input is either active now or has never become active since the RIDLE flag was last cleared
- 1 = Receiver input has become idle (after receiving a valid fame)

Note: When the Receiver Wake Up (RWU) bit is set, an idle line condition *does not set the RIDLE flag.*

10.9.4.5 Overrun Flag (OR)—Bit 11

This bit is set when software fails to read the SCI Data Register (SCIDR) before the Receive Shift register receives the next frame. The data in the Shift register is lost, but the data already in the SCI Data register is not affected. Clear OR by reading the SCI Status Register (SCISR) with OR set, then write the SCISR with any value.

- 0 = No overrun
- 1 = Overrun

10.9.4.6 Noise Flag (NF)—Bit 10

This bit is set when the SCI detects noise on the receiver input. The NF bit is set during the same cycle as the RDRF flag, but it is not set in the case of an overrun. Clear NF by reading the SCISR then write the SCI Status register with any value.

- 0 = No noise
- 1 = Noise

10.9.4.7 Framing Error Flag (FE)—Bit 9

This bit is set when a Logic 0 is accepted as the Stop bit. FE bit is set during the same cycle as the RDRF flag but it is not set in the case of an overrun. FE inhibits further data reception until it is cleared. Clear FE by reading the SCISR with FE set, then write the

SCISR with any value.

- 0 = No framing error
- 1 = Framing error

10.9.4.8 Parity Error Flag (PF)—Bit 8

This bit is set when the Parity Enable (PE) bit is set and the parity of the received data does not match its parity bit. Clear PF by reading the SCISR then write the SCISR with any value.

- 0 = No parity error
- 1 = Parity error

10.9.4.9 Reserved—Bits 7–1

These bits are reserved or not implemented. They are read as 0, but cannot be modified by writing.

10.9.4.10 Receiver Active Flag (RAF)—Bit 0

This bit is set when the receiver detects a Logic 0 during the RT1 time period of the Start bit search. RAF is cleared when the receiver detects false Start bits (usually from noise or baud rate mismatch) or when the receiver detects a preamble.

- 0 = No reception in progress
- 1 = Reception in progress

10.9.5 SCI Data Register (SCIDR)

The SCI Data Register (SCIDR) can be read and modified at any time. Reading accesses SCI Receive Data Register (SRDR). Writing to the register accesses SCI Transmit Data Register (STDR).

Base + $4	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	0	0	0	0	0	0	0	RECEIVE DATA								
Write								TRANSMIT DATA								
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 10-21. SCI Data Register (SCIDR)

See Programmer's Sheets on Appendix page B-61

10.9.5.1 Reserved—Bits 15–9

These bits are reserved or not implemented. They are read and written as 0.

10.9.5.2 Receive Data—Bits 8–0

Data received.

10.9.5.3 Transmit Data—Bits 8–0

Data to be transmitted.

10.10 Clocks

All timing is derived from the IPBus clock at half of the system clock for this module. Please see **Section 10.6.2** for a description of how the data rate is determined.

10.11 Resets

Reset characteristics are determined by the state of control register bit settings. Therefore, the register descriptions comprising **Section 10.9** cover all reset functions.

10.12 Interrupts

Table 10-12. SCI Interrupt Sources

Interrupt Source	Flag	Local Enable
Transmitter	TDRE	TEIE
Transmitter	TIDLE	TIIE
Receiver	RDRF	RFIE
	OR	
Receiver	FE	REIE
	PE	
	NF	
	OR	
Receiver	RIDLE	RIIE

10.12.1 Transmitter Empty Interrupt

This interrupt is enabled by setting the TEIE bit of the SCICR. When this interrupt is enabled an interrupt is generated when data is transferred from the SCI Data register to the Transmit Shift register. The interrupt service routine should read the SCISR and verify the TDRE bit is set, and then write the next data to be transmitted to the SCIDR, clearing the TDRE bit.

10.12.2 Transmitter Idle Interrupt

This interrupt is enabled by setting the TIIE bit of the SCICR. This interrupt indicates the TDRE flag is set and the transmitter is no longer sending data, preamble, or break characters. The interrupt service routine should read the SCISR, verifying the TIDLE bit is set, then initiate a preamble, break, or write a data character to the SCIDR. Any of these actions will clear the TIDLE bit because the transmitter is busy at that time.

10.12.3 Receiver Full Interrupt

This interrupt is enabled by setting the RIE bit of the SCICR. This interrupt indicates the receive data is available in the SCIDR. The interrupt service routine should read the SCISR, verifying the RDRF bit is set, then read the data from the SCISR. This will clear the RDRF bit.

10.12.4 Receive Error Interrupt

This interrupt is enabled by setting the REIE bit of the SCICR. This interrupt indicates any of the listed errors were detected by the receiver:

1. Noise Flag (NF) set
2. Parity error Flag (PF) set

3. Framing Error (FE) flag set
4. OverRun (OR) flag set

The interrupt service routine should read the SCISR to determine which of the error flags was set. The error flag is set by writing (anything) to the SCISR. The appropriate action should then be taken by the software to handle the error condition.

10.12.5 Receiver Idle Interrupt

This interrupt is used in conjunction with receive DMA operation. Please see the RIIE bit description in **Section 10.7.3** and **Section 10.9.3** for a description of the intended operation of this interrupt. When this interrupt occurs the appropriate response of the interrupt service routine would be to disable the RIIE until the next message receive sequence occurs.

Chapter 11
Serial Peripheral Interface (SPI)

11.1 Introduction

This section describes the Serial Peripheral Interface (SPI) module available on each chip, with the exception for the 56855. The module allows full-duplex, synchronous, serial communication between the Digital Signal Controllers (DSC) and peripheral devices, including other DSCs. Software can poll SPI status flags or SPI operation can be interrupt driven. This block contains four 16-bit memory mapped registers for control parameters, status, and data transfer.

11.2 Features

Features of the SPI module include:

- Full-duplex operation
- Master and slave modes
- Double-buffered operation with separate transmit and receive registers
- Programmable length transmissions (2 to 16 bits)
- Programmable transmit and receive shift order (MSB first or last bit transmitted)
- Eight master mode frequencies (maximum = bus frequency/2^1)
- Maximum slave mode frequency = bus frequency
- Clock ground for reduced Radio Frequency (RF) interference
- Serial clock with programmable polarity and phase
- Two separately enabled interrupts
  - SPRF (SPI Receiver Full)
  - SPTE (SPI Transmitter Empty)
- Mode Fault Error flag with interrupt capability
- Direct Memory Access (DMA) capability for both the transmitter and receiver
- Wired OR mode functionality to enabling connection to multiple SPIs
- Overflow Error Flag with interrupt capability

1. This frequency may be further limited by the GPIO function.

11.3 SPI Block Diagram

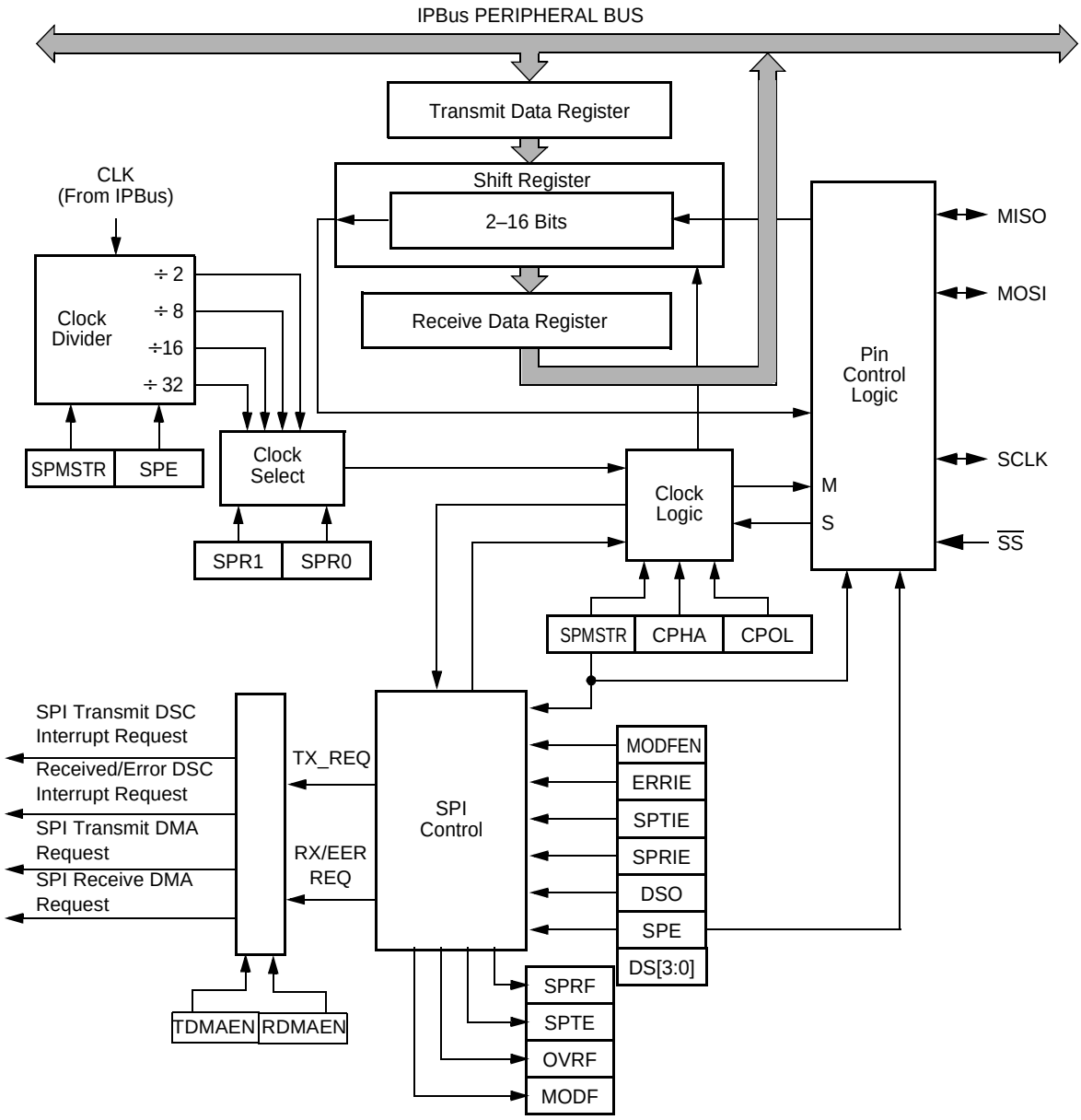


Figure 11-1. SPI Block Diagram

11.4 Signal Descriptions

11.4.1 Master In/Slave Out (MISO)

MISO is one of the two SPI module pins to transmit serial data. In full duplex operation, the MISO pin of the Master SPI module is connected to the MISO pin of the slave SPI module. The master SPI simultaneously receives data on its MISO pin and transmits data from its MOSI pin.

Slave output data on the MISO pin is enabled only when the SPI is configured as a slave. The SPI is configured as a slave when the SPMSTR bit, illustrated in **Table 11-4,** is Logic 0 and its $\overline{SS}$ pin is at Logic 0. To support a multiple slave system, a Logic 1 on the $\overline{SS}$ pin puts the MISO pin in a high impedance state.

11.4.2 Master Out/Slave In (MOSI)

MOSI is one of the two SPI module pins to transmit serial data. In full duplex operation, the MOSI pin of the Master SPI module is connected to the MOSI pin of the Slave SPI module. The Master SPI simultaneously transmits data from its MOSI pin and receives data on its MISO pin.

11.4.3 Serial Clock (SCLK)

The serial clock synchronizes data transmission between master and slave devices. In a Master DSC, the SCLK pin is the clock output. In a slave DSC, the SCLK pin is the clock input. In full duplex operation, the master and slave exchange data in the same number of clock cycles as the number of bits of transmitted data.

11.4.4 Slave Select ($\overline{SS}$)

The $\overline{SS}$ pin has various functions depending on the current state of the SPI. For a SPI configured as a slave, the $\overline{SS}$ is used to select a slave. For CPHA = 0, the $\overline{SS}$ is used to define the start of a transmission. Since it is used to indicate the start of a transmission, the $\overline{SS}$ must be toggled high and low between each full length data transmitted for the CPHA = 0 format. However, it can remain low between transmissions for the CPHA = 1 format. Please refer to **Figure 11-2.**

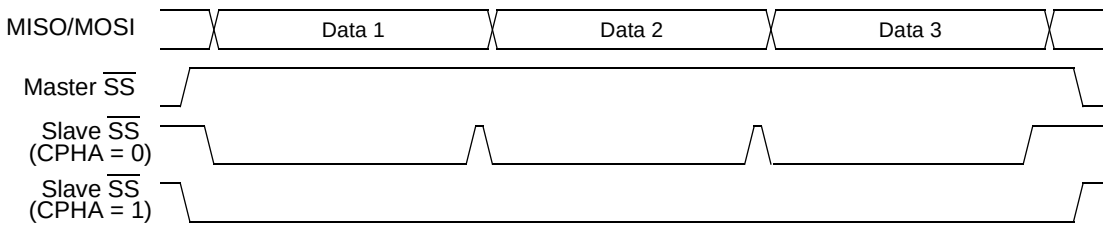


Figure 11-2. CPHA/$\overline{SS}$ Timing

When an SPI is configured as a slave, the $\overline{SS}$ pin is always configured as an input. The MODFEN bit can prevent the state of the $\overline{SS}$ from creating a MODF error.

Note: A Logic 1 voltage on the $\overline{SS}$ pin of a slave SPI puts the MISO pin in a high-impedance state. The slave SPI ignores all incoming SCLK clocks, even if it was already in the middle of a transmission. A Mode Fault will occur if the $\overline{SS}$ pin changes state during a transmission.

When a SPI is configured as a master, the $\overline{SS}$ input can be used in conjunction with the MODF flag to prevent multiple masters from driving MOSI and SCLK. For the state of the $\overline{SS}$ pin to set the MODF flag, the MODFEN bit in the SCLK register must be set.

Table 11-1. SPI I/O Configuration

SPE	SPMSTR	MODFEN	SPI Configuration	State of $\overline{SS}$ Logic
0	X[1]	X	Not Enabled	$\overline{SS}$ ignored by SPI
1	0	X	Slave	Input-only to SPI
1	1	0	Master without MODF	$\overline{SS}$ ignored by SPI
1	1	1	Master with MODF	Input-only to SPI

1. X = Don't care

11.5 External I/O Signals

There are four external SPI pins. All are summarized in **Table 11-2.**

Table 11-2. External I/O Signals

Signal Name	Description	Direction
MOSI	Master-Out Slave-In Pad Pin	Bi-Directional
MISO	Master-In Slave-Out Pad Pin	Bi-Directional
SCLK	Slack Clock Pad Pin	Bi-Directional
$\overline{SS}$	Slave Select Pad Pin (Active Low)	Input

11.6 Operating Modes

The SPI has two operating modes:

- Master
- Slave

An operating mode is selected by the SPMSTR bit in the SPSCR as follows:

- SPMSTR = 0 Slave mode
- SPMSTR = 1 Master mode

Note: The SPMSTR bit should be configured before enabling the SPI, setting the SPE bit in the SPSCR. The master SPI should be enabled before enabling any slave SPI. All slave SPIs should be disabled before disabling the master SPI.

11.6.1 Master Mode

The SPI operates in Master mode when the SPI master bit, SPMSTR, is set.

Note: Configure the SPI module as master or slave before enabling the SPI. Enable the master SPI before enabling the slave SPI. Disable the slave SPI before disabling the master SPI.

Only a Master SPI module can initiate transmissions. With the SPI enabled, software begins the transmission from the Master SPI module by writing to the SPI Data Transmit Register (SPDTR). If the Shift register is empty, the data immediately transfers to the Shift register, setting the SPI Transmitter Empty (SPTE) bit. The data begins shifting out on the MOSI pin under the control of the SPI Serial Clock (SCLK).

The SPR2, SPR1 and SPR0 bits in the SPSCR register, control the baud rate generator and determine the speed of the Shift register. Through the SCLK pin, the baud rate generator of the master also controls the Shift register of the slave peripheral.

As the data shifts out on the MOSI pin of the master, external data shifts in from the slave on the master's MISO pin. The transmission ends when the Receiver Full (SPRF) bit becomes set. At the same time the SPRF becomes set the data from the slave transfer to the Receive Data register. In normal operation, SPRF signals the end of a transmission. Software clears SPRF by reading the SPI Data Receive Register (SPDRR). Writing to the SPI Data Transmit Register (SPDTR) clears the SPTE bit.

Figure 11-3 is an example configuration for a Full-Duplex Master-Slave Configuration. Having the $\overline{SS}$ bit of the Master DSC held high is only necessary if MODFEN = 1. Tying the Slave DSC $\overline{SS}$ bit to ground should only be executed if CPHA = 1.

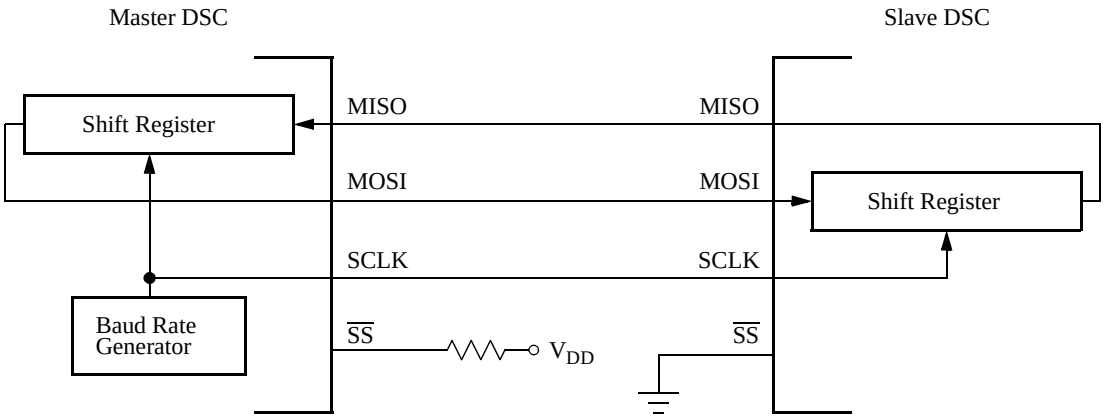


Figure 11-3. Full-Duplex Master/Slave Connections

11.6.2 Slave Mode

The SPI operates in the Slave mode when the SPMSTR bit is clear. While in the Slave mode, the SCLK pin is the input for the serial clock from the master DSC. Before a data transmission occurs, the $\overline{SS}$ pin of the slave SPI must be at Logic 0. $\overline{SS}$ must remain low until the transmission is complete or a Mode Fault Error occurs.

Note: The SPI must be enabled (SPE = 1) for slave transmissions to be received.

Note: Data in the Transmitter Shift register will be unaffected by SCLK transitions in the event the SPI is operating as a slave but is deselected.

In a slave SPI module, data enters the Shift register under the control of the Serial Clock (SCLK) from the Master SPI module. After a full length data transmission enters the Shift register of a slave SPI, it transfers to the SPI Data Receive Register (SPDRR) and the SPI Receiver Full (SPRF) bit in the SPI Status and Control Register (SPSCR) is set. If the Receive Data Register (SPDRR), and the SPRF bit are set in the SPSCR, a receive interrupt is also generated. To prevent an overflow condition, slave software must read the SPDRR before another full length data transmission enters the Shift register.

The maximum frequency of the SCLK for a SPI configured as a slave is the bus clock speed. The bus clock speed is twice as fast as the fastest master SCLK potential generation. Frequency of the SCLK for an SPI configured as a slave does not have to correspond to any particular SPI baud rate. The baud rate only controls the speed of the SCLK generated by an SPI configured as a master. Therefore, the frequency of the SCLK for a SPI configured as a slave can be any frequency less than, or equal to, the bus speed.

When the master SPI starts a transmission, the data in the slave Shift register begins shifting out on the MISO pin. The slave can load its Shift register with new data for the next transmission by writing to its Transmit Data register. The slave must write to its Transmit Data register at least one bus cycle before the master begins the next transmission. Otherwise, the data already in the slave Shift register shifts out on the MISO pin. Data written to the Slave Shift register during a transmission remains in a buffer until the end of the transmission.

When the CPHA bit is set, the first edge of SCLK starts a transmission. When CPHA is cleared, the falling edge of $\overline{SS}$ starts a transmission.

Note: SCLK must be in the proper idle state before the slave is enabled to prevent SCLK from appearing as a clock edge.

11.6.3 DMA Mode

When the TDMAEN or RDMAEM bits are set, the SPI operates in DMA mode. Normal SPTE and/or SPRF interrupts are suppressed. Instead DMA requests are generated, signaling the DMA controllers to read the SPDRR or SPDTR as needed. The DMA requests are suppressed in the case of an OVRF or MODF interrupt, illustrated in **Figure 11-4.** If the SPI is being used to send and receive data, both TDMAEN and RDMAEN should be enabled/disabled at the same time. If data is only being received or transmitted TDMAEN or RDMAEN may be left inactive as appropriate, however in this case ERRIE must not be set to prevent the DMA request being masked by a mode-fault error or overflow error. DMA mode must be used carefully. In the case where CPHA=0, the $\overline{SS}$ signal to the slave is toggled by software between words. In DMA mode there is no way of generating this pulse easily, so the DMA should not be used when CPHA=0.

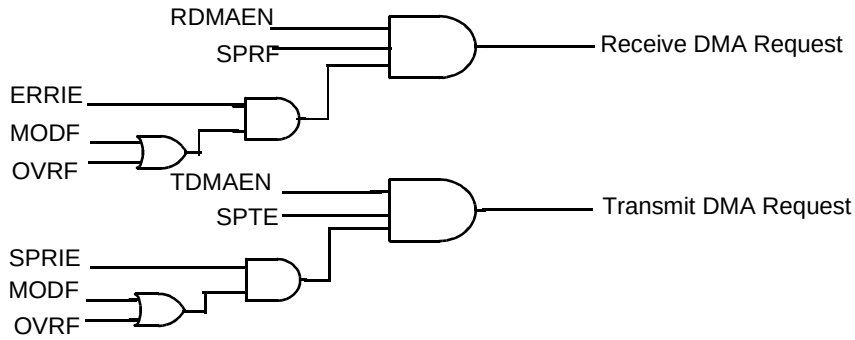


Figure 11-4. SPI DMA Request Generation

11.6.4 Wired OR Mode

Wired OR functionality is provided to permit the connection of multiple SPIs. **Figure 11-5** illustrates a single master controlling multiple slave SPIs. When the WOM bit is set, the outputs switch from conventional complementary CMOS output to open drain outputs.

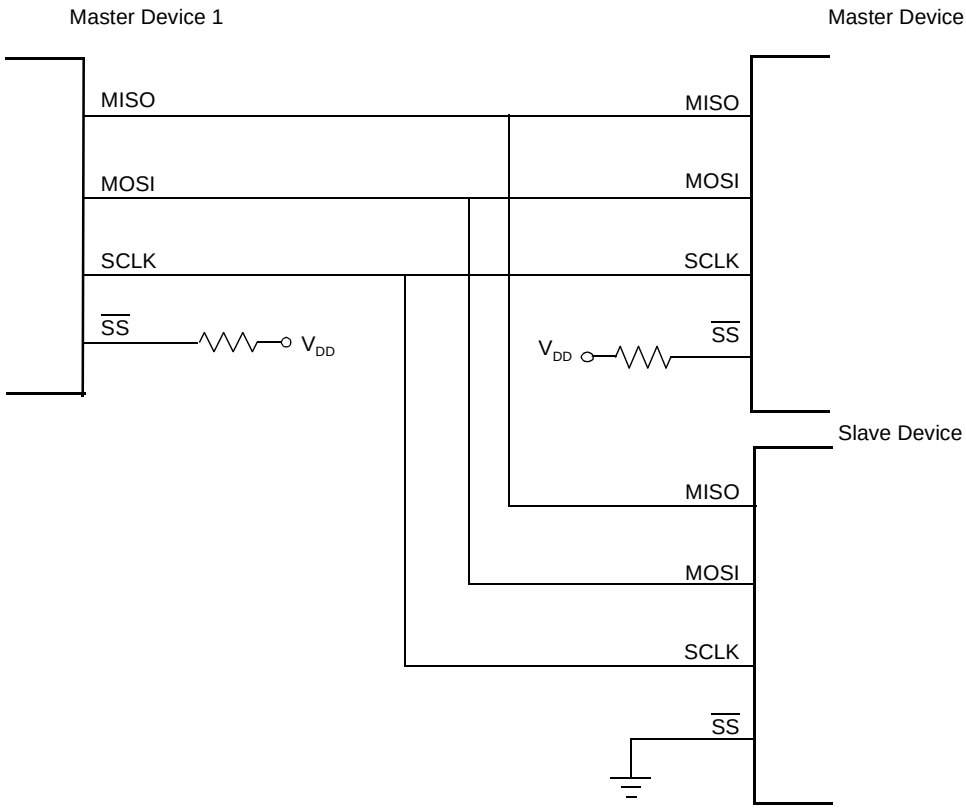


Figure 11-5. Sharing of a Slave by Multiple Masters

This lets the internal pull-up resistor bring the line high and whichever SPI drives the line pulls it low as needed.

11.7 Transmission Formats

During a SPI transmission, data is simultaneously transmitted, or shifted out serially, and received; that is it is shifted in serially. A serial clock synchronizes shifting and sampling on the two serial data lines. A Slave Select line allows selection of an individual slave SPI device; slave devices not selected do not interfere with SPI bus activities. On a master SPI device, the Slave Select line can optionally be used to indicate multiple-master bus contention.

11.7.1 Data Transmission Length

The SPI can support data lengths from one to 16 bits. This can be configured in the Data Size Register (SPDSR). When the data length is less than 16 bits, the Receive Data register will pad the upper bits with zeros. It is the responsibility of the software to remove these upper bits since 16 bits will be read when reading the Receive Data register (SPDRR).

Note: Data can be lost if the data length is not the same for both master and slave devices.

11.7.2 Data Shift Ordering

The SPI can be configured to transmit or receive the MSB of the desired data first or last. This is controlled by the Data Shift Order (DSO) bit in the SPSCR. Regardless which bit is transmitted or received first, the data shall always be written to the SPI Data Transmit Register (SPDTR) and read from the Receive Data Register (SPDRR) with the LSB in bit 0 and the MSB in the correct position, depending on the data transmission size.

11.7.3 Clock Phase and Polarity Controls

Software can select any of four combinations of Serial Clock (SCLK) phase and polarity using two bits in the SPI Status and Control Register (SPSCR). The Clock Polarity is specified by the (CPOL) control bit. In turn, it selects an active high or low clock. It has no significant effect on the transmission format.

The Clock Phase (CPHA) control bit selects one of two fundamentally different transmission formats. The clock phase and polarity should be identical for the master SPI device and the communicating slave device. In some cases, the phase and polarity are changed between transmissions, allowing a master device to communicate with peripheral slaves with different requirements.

Note: Before writing to the CPOL bit or the CPHA bit, disable the SPI by clearing the SPI Enable (SPE) bit. Do not change SPE and CPHA or CPOL at the same time.

11.7.4 Transmission Format When CPHA = 0

Figure 11-6 exhibits a SPI transmission with CPHA as Logic 0. The figure should not be used as a replacement for data sheet parametric information. Two waveforms are shown for SCLK:

1. CPOL = 0
2. CPOL = 1

The diagram may be interpreted as a master or slave timing diagram since the Serial Clock (SCLK), Master In/Slave Out (MISO), and Master Out/Slave In (MOSI) pins are directly

connected between the master and the slave. The MISO signal is the output from the slave, and the MOSI signal is the output from the master.

The $\overline{SS}$ line is the slave select input to the slave. The slave SPI drives its MISO output only when its Slave Select input ($\overline{SS}$) is at Logic 0 because only the selected slave drives to the master. The $\overline{SS}$ pin of the master is not shown, but it is assumed to be inactive. While MODFEN = 1, the $\overline{SS}$ pin of the master must be high or a Mode Fault Error will occur. If MODFEN = 0, the state of the $\overline{SS}$ is ignored. When CPHA = 0, the first SCLK edge is the MSB capture strobe. Therefore, the slave must begin driving its data before the first SCLK edge and a falling edge on the $\overline{SS}$ pin is used to start the slave data transmission. The slave's $\overline{SS}$ pin must be toggled back to high and then low again between each full length data transmitted as depicted in **Figure 11-7.**

Note: **Figure 11-6** assumes 16-bit data lengths and the MSB shifted out first.

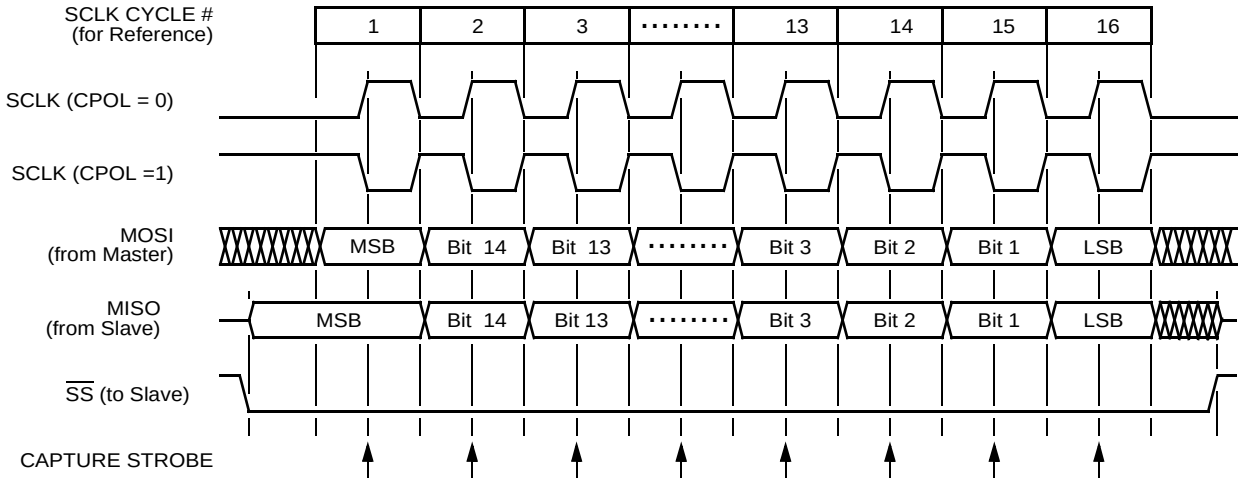


Figure 11-6. Transmission Format (CPHA = 0)

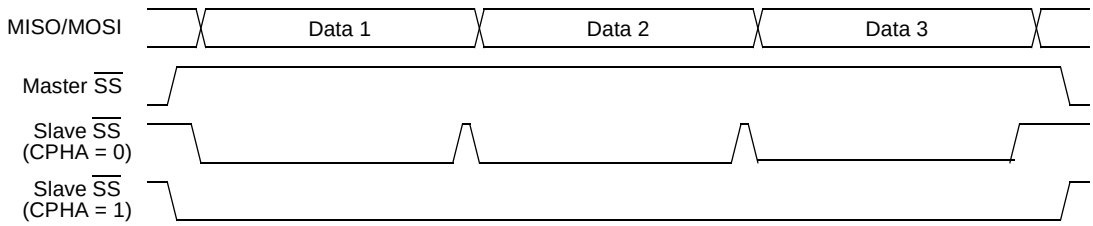


Figure 11-7. CPHA/$\overline{SS}$ Timing

When CPHA = 0 for a slave, the falling edge of $\overline{SS}$ indicates the beginning of the transmission. This causes the SPI to leave its idle state and begin driving the MISO pin with the first bit of its data. Once the transmission begins, no new data is allowed into the Shift register from the SPI Data Transmit Register (SPDTR). Therefore, the SPDTR of the slave must be loaded with

transmit data before the falling edge of $\overline{SS}$. Any data written after the falling edge is stored in the SPDTR and transferred to the Shift register after the current transmission.

When CPHA = 0 for a master, normal operation would begin by the master initializing the $\overline{SS}$ pin of the slave high. A transfer would then begin by the master setting the $\overline{SS}$ pin of the slave low and then writing the SPDTR. After completion of a data transfer, the $\overline{SS}$ pin would be put back into the high state by the master device.

11.7.5 Transmission Format When CPHA = 1

A SPI transmission is shown in **Figure 11-8** where CPHA is Logic1. The figure should not be used as a replacement for data sheet parametric information. Two waveforms are shown for SCLK: 1 for CPOL = 0 and another for CPOL = 1. The diagram may be interpreted as a master or slave timing diagram since the serial clock (SCLK), Master In/Slave Out (MISO), and Master Out/Slave In (MOSI) pins are directly connected between the master and the slave. The MISO signal is the output from the slave, and the MOSI signal is the output from the master. The $\overline{SS}$ line is the Slave Select input to the slave. The slave SPI drives its MISO output only when its Slave Select input ($\overline{SS}$) is at Logic 0, so only the selected slave drives to the master. The $\overline{SS}$ pin of the master is not shown but is assumed to be inactive. The $\overline{SS}$ pin of the master must be high or a Mode Fault Error occurs. When CPHA = 1, the master begins driving its MOSI pin on the first SCLK edge. Therefore, the slave uses the first SCLK edge as a start transmission signal. The $\overline{SS}$ pin can remain low between transmissions. This format may be preferable in systems having only one master and slave driving the MISO data line.

Note: **Figure 11-8** assumes 16-bit data lengths and the MSB shifted out first.

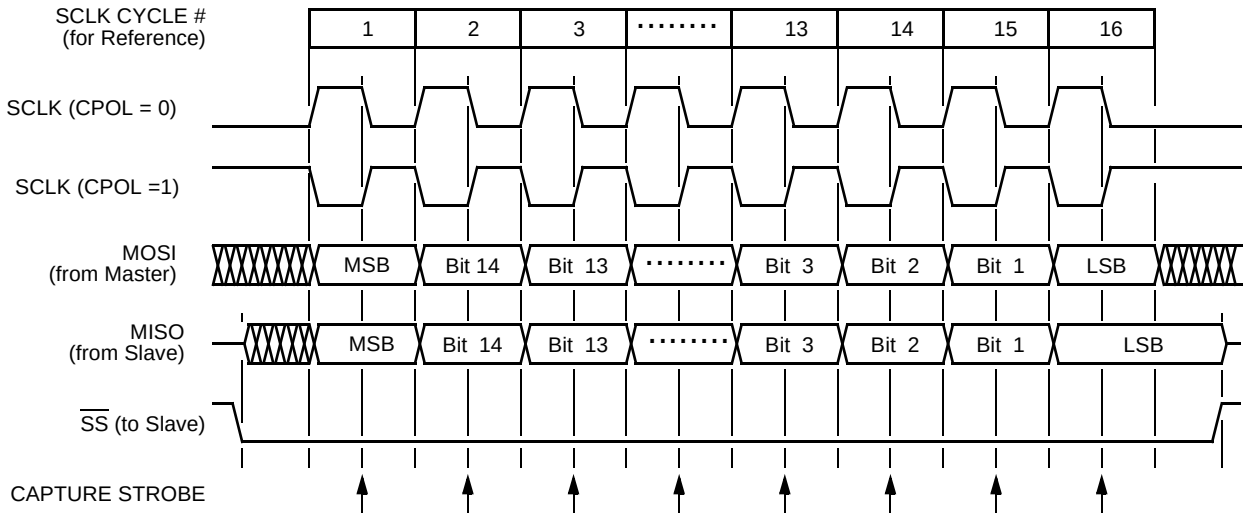


Figure 11-8. Transmission Format (CPHA = 1)

When CPHA = 1 for a slave, the first edge of the SCLK indicates the beginning of the transmission. This causes the SPI to leave its idle state and begin driving the MISO pin with the first bit of its data. Once the transmission begins, no new data is allowed into the Shift Register from the Data Transmit Register. Therefore, the SPI Data Register of the slave must be loaded with transmit data before the first edge of SCLK. Any data written after the first edge is stored in the Data Transmit Register and transferred to the Shift Register after the current transmission.

11.7.6 Transmission Initiation Latency

When the SPI is configured as a master (SPMSTR = 1), writing to the SPDTR starts a transmission. CPHA has no effect on the delay to the start of the transmission, but it does affect the initial state of the SCLK signal. When CPHA = 0, the SCLK signal remains inactive for the first half of the first SCLK cycle. When CPHA = 1, the first SCLK cycle begins with an edge on the SCLK line from its inactive to its active level. The SPI clock rate, selected by SPR1:SPR0, affects the delay from the write to SPDTR and the start of the SPI transmission. The internal SPI clock in the master is a free-running derivative of the internal clock. To conserve power, it is enabled only when both the SPE and SPMSTR bits are set. Since the SPI clock is free-running, it is uncertain where the write to the SPDTR occurs relative to the slower SCLK. This uncertainty causes the variation in the initiation delay, demonstrated in **Figure 11-9.** This delay is no longer than a single SPI bit time. That is, the maximum delay is two bus cycles for DIV2, four bus cycles for DIV4, eight bus cycles for DIV8, and so on up to a maximum of 256 cycles for DIV256.

Note: **Figure 11-9** assumes 16-bit data lengths and the MSB shifted out first.

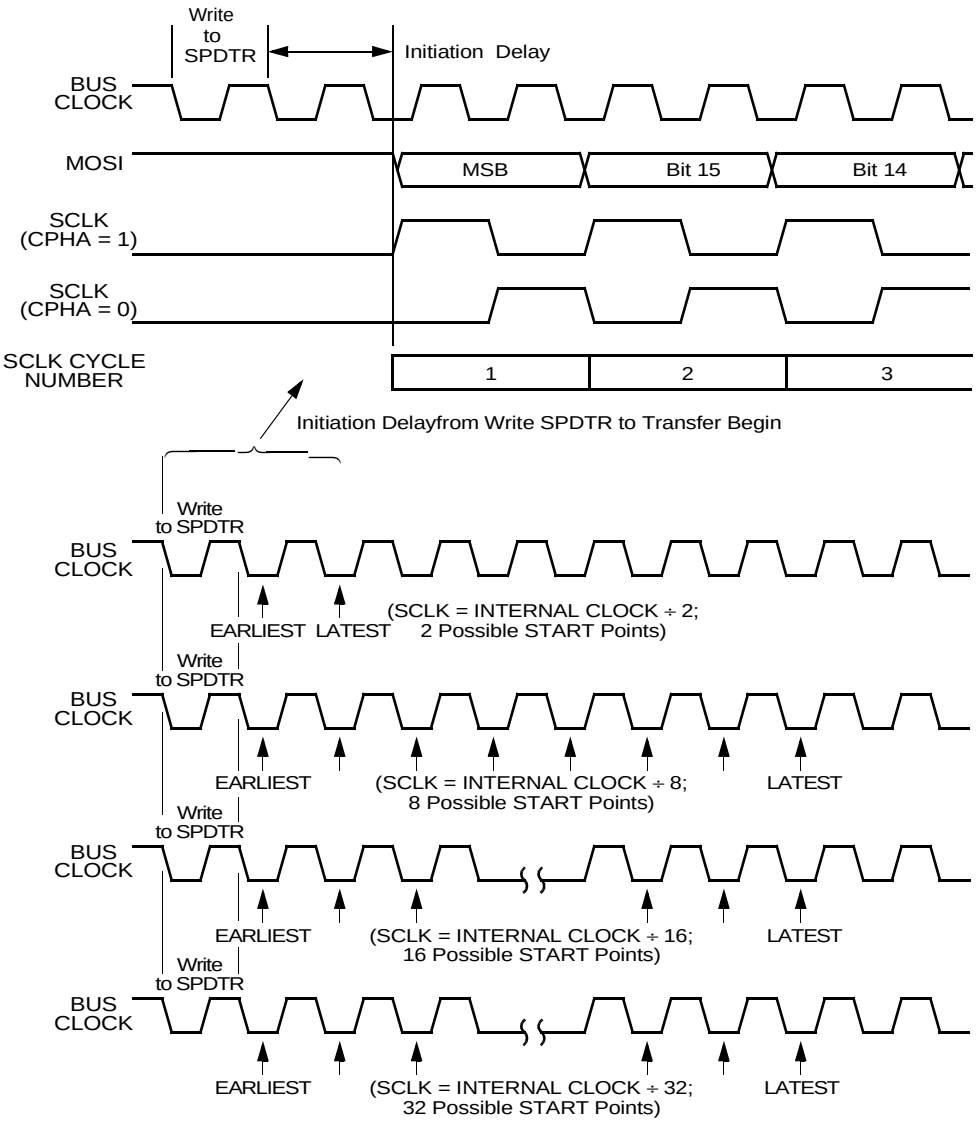
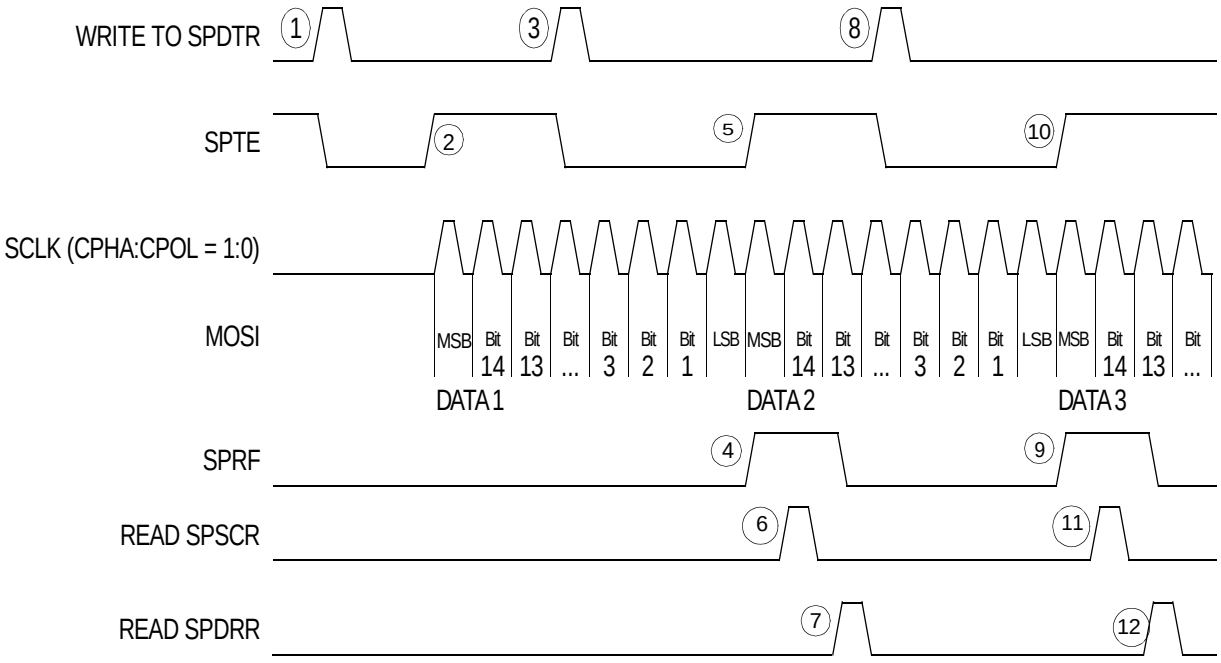


Figure 11-9. Transmission Start Delay (Master)

11.8 Transmission Data

The double-buffered Data Transmit Register (SPDTR) allows data to be queued and transmitted. For a SPI configured as a master, the queued data is transmitted immediately after the previous transmission has completed. The SPI Transmitter Empty (SPTE) flag indicates when the transmit data buffer is ready to accept new data. Write to the SPDTR only when the SPTE bit is high. **Figure 11-10** illustrates the timing associated with doing back-to-back transmissions with the SPI (SCLK has CPHA: CPOL = 1:0).

Note: **Figure 11-10** assumes 16-bit data lengths and the MSB shifted out first.



① DSC WRITES DATA 1 TO SPDTR, CLEARING THE SPTE BIT.

② DATA 1TRANSFERS FROM TRANSMIT DATA REGISTER TO SHIFT REGISTER, SETTING SPTE BIT.

③ DSC WRITES DATA 2 TO SPDTR, QUEUEING DATA 2 AND CLEARING SPTE BIT.

④ FIRST INCOMING WORD TRANSFERS FROM THE SHIFT REGISTER TO THE RECEIVE DATA REGISTER, SETTING THE SPRF BIT.

⑤ DATA 2 TRANSFERS FROM THE TRANSMIT DATA REGISTER TO SHIFT REGISTER, SETTING THE SPTE BIT.

⑥ DSC READS SPSCR WITH THE SPRF BIT SET.

⑦ DSC READS SPDRR, CLEARING SPRF BIT.

⑧ DSC WRITES DATA 3 TO SPDTR, QUEUEING DATA 3 AND CLEARING THE SPTE BIT.

⑨ SECOND INCOMING DATA TRANSFERS FROM SHIFT REGISTER TO RECEIVE DATA REGISTER SETTING THE SPRF BIT.

⑩ DATA 3 TRANSFERSF ROM THE TRANSMIT DATA REGISTER TO THE SHIFT REGISTER, SETTING THE SPTE BIT.

⑪ DSC READS SPSCR WITH THE SPRF BIT.

⑫ DSC READS SPDRR, CLEARING THE SPRF BIT.

Figure 11-10. SPRF/SPTE Interrupt Timing

The transmit data buffer permits back-to-back transmissions without the slave precisely timing its writes between transmissions as is necessary in a system with a single data buffer. Also, if no new data is written to the data buffer, the last value contained in the Shift register is the next data to be transmitted.

An idle master or idle slave without loaded data in its transmit buffer, sets the SPTE again no more than two bus cycles after the transmit buffer empties into the Shift register. This allows a queue to send up to a 32-bit value. For an already active slave, the load of the Shift register

cannot occur until the transmission is completed. This implies a back-to- back write to the Data Transmit Register (SPDTR) is not possible. The SPTE indicates when the next write can occur.

11.9 Error Conditions

The following flags signal SPI error conditions:

- Overflow (OVRF) — Failing to read the SPI Data Register before the next full length data enters the Shift register sets the OVRF bit. The new data will not transfer to the Data Receive Register (SPDRR), and the unread data can still be read. OVRF is in the SPI Status and Control Register (SPSCR).
- Mode Fault Error (MODF) — The MODF bit indicates the voltage on the Slave Select pin ($\overline{SS}$) is inconsistent with the mode of the SPI. MODF is in the SPSCR.

11.9.1 Overflow Error

The Overflow Flag (OVRF) becomes set if the Data Receive Register (SPDRR) still has unread data from a previous transmission and when bit one's capture strobe of the next transmission occurs. Bit 1 capture strobe occurs in the middle of SCLK when the Data Length = Transmission Data Length - 1. If an overflow occurs, all data received after the overflow, and before the OVRF bit is cleared, does not transfer to the SPDRR. It does not set the SPI Receiver Full (SPRF) bit. The unread data transferred to the SPDRR before the overflow occurred can still be read. Therefore, an overflow error always indicates the loss of data. Clear the overflow flag by reading the SPI Status and Control Register (SPSCR), then read the SPI Data register.

OVRF generates a receiver/error interrupt request if the Error Interrupt Enable (ERRIE) bit is also set. It is not possible to enable MODF or OVRF individually in order to generate a receiver/error interrupt request. However, leaving MODFEN low prevents MODF from being set.

If the SPRF interrupt is enabled and the OVRF interrupt is not, watch for an overflow condition. **Figure 11-11** explains how it is possible to miss an overflow. The first element of the same figure illustrates how it is possible to read the SPSCR and SPDRR to clear the SPRF without problems. However, as illustrated by the second transmission example, the OVRF bit can be set between the time SPSCR and SPDRR are read.

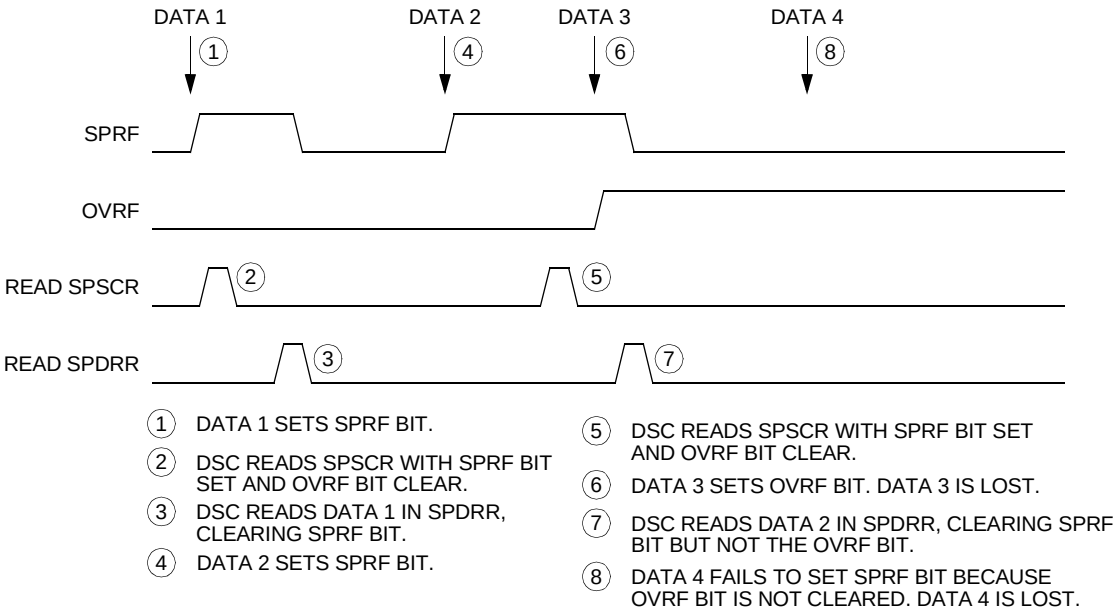


Figure 11-11. Missed Read of Overflow Condition

In this case, an overflow can easily be missed. Since no more SPRF interrupts can be generated until this OVRF is serviced, it is not obvious data is being lost as more transmissions are completed. To prevent this loss, either enable the OVRF interrupt or take another read of the SPSCR following the read of the SPDRR. This ensures the OVRF was not set before the SPRF was cleared. Future transmissions can set the SPRF bit. **Figure 11-11** illustrates the described process. Generally, to avoid a second SPSCR read, enable the OVRF to the core by setting the ERRIE bit.

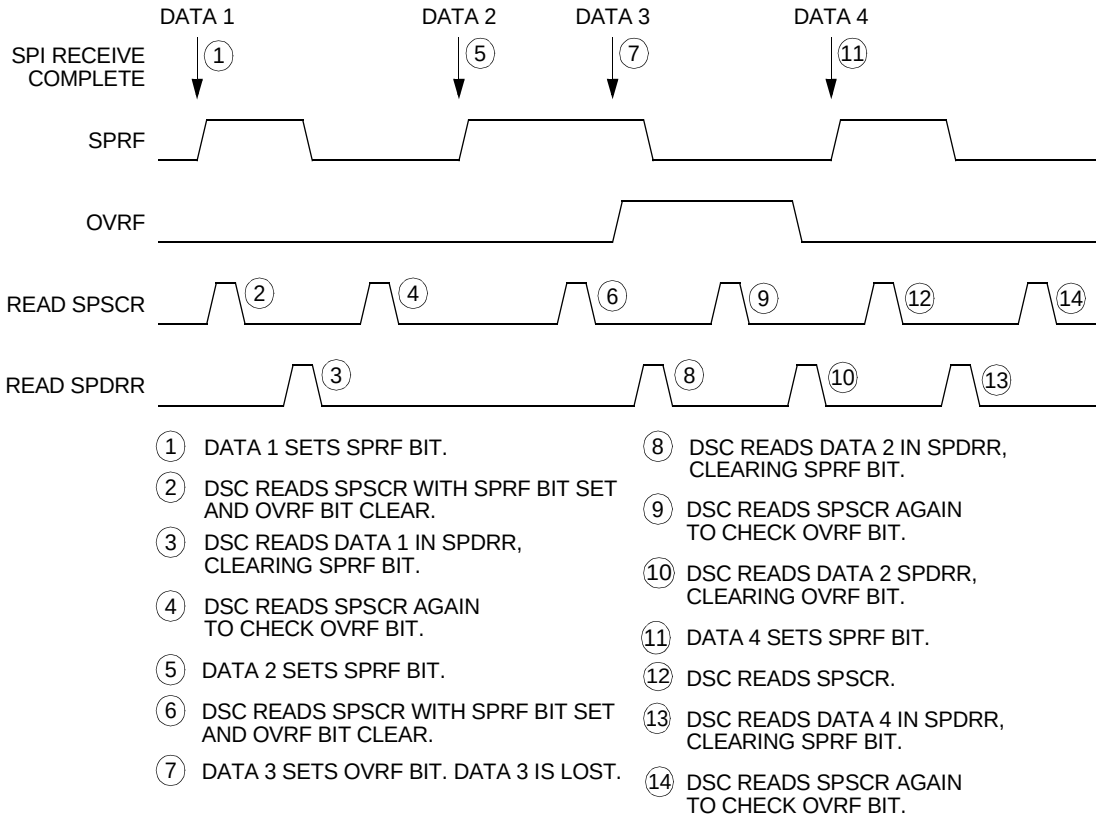


Figure 11-12. Clearing SPRF When OVRF Interrupt is Not Enabled

11.9.2 Mode Fault Error

Setting the SPMSTR bit selects the Master mode, configuring the SCLK and MOSI pins as outputs and the MISO pin as an input. Clearing SPMSTR selects the Slave mode, configuring the SCLK and MOSI pins as inputs and the MISO pin as an output. The Mode Fault (MODF) bit becomes set any time the state of the Slave Select ($\overline{SS}$) pin is inconsistent with the mode selected by SPMSTR. To prevent SPI pin contention and damage to the DSC, a Mode Fault Error occurs if:

- The $\overline{SS}$ pin of a Slave SPI goes high during a transmission.
- The $\overline{SS}$ pin of a Master SPI goes low at any time.

To set the MODF flag, the Mode Fault Error Enable (MODFEN) bit must be set. Clearing the MODFEN bit does not clear the MODF flag, but it does prevent the MODF from being set again after the MODF is cleared.

MODF generates a Receiver/Error Interrupt request if the Error Interrupt Enable (ERRIE) bit is also set. It is not possible to enable MODF or OVRF individually to generate a Receiver/Error Interrupt request. However, leaving MODFEN low prevents MODF from being set.

In a Master SPI with the Mode Fault Enable (MODFEN) bit set, the Mode Fault (MODF) flag is set if $\overline{SS}$ goes to Logic 0. A Mode Fault in a Master SPI causes the following events to occur:

- If ERRIE = 1, the SPI generates a SPI receiver/error interrupt request
- The SPE bit is cleared (SPI disabled)
- The SPTE bit is set
- The SPI state counter is cleared

When configured as a slave (SPMSTR = 0), the MODF flag is set if the $\overline{SS}$ goes high during a transmission. When CPHA = 0, a transmission begins when $\overline{SS}$ goes low and ends once the incoming SCLK goes back to its idle level, following the shift of the last data bit. When CPHA = 1, the transmission begins when the SCLK leaves its idle level and $\overline{SS}$ is already low. The transmission continues until the SCLK returns to its idle level following the shift of the last data bit.

Note: Setting the MODF flag does not clear the SPMSTR bit. The SPMSTR bit has no function when SPE = 0. Reading SPMSTR when MODF = 1 shows the difference between a MODF occurring when the SPI is a master and when it is a slave.

When CPHA = 0, a MODF occurs if a slave is selected ($\overline{SS}$ is at Logic 0) and later unselected ($\overline{SS}$ is at Logic 1) even if no SCLK is sent to that slave. This happens because $\overline{SS}$ at Logic 0 indicates the start of the transmission (MISO driven out with the value of MSB) for CPHA = 0. When CPHA = 1, a slave can be selected and then later unselected with no transmission occurring. Therefore, MODF does not occur since a transmission was never begun.

In a Slave SPI (MSTR = 0), the MODF bit generates a SPI Receiver/Error Interrupt request if the ERRIE bit is set. The MODF bit does not clear the SPE bit or reset the SPI in any way. Software can abort the SPI transmission by clearing the SPE bit of the slave.

Note: A logic one voltage on the $\overline{SS}$ pin of a slave SPI puts the MISO pin in a high impedance state. Also, the slave SPI ignores all incoming SCLK clocks, even if it was already in the middle of a transmission.

In a master SPI, the MODF flag will not be cleared until the $\overline{SS}$ pin is at a Logic 1 or the SPI is configured as a slave.

In a slave SPI, if the MODF flag is not cleared by writing 1 to the MODF bit, the condition causing the Mode Fault still exists. The MODF flag and corresponding interrupt can be cleared by disabling the EERIE or MODFEN bits (if set) or by disabling the SPI. It is possible to clear the MODF error condition by disabling the SPE or MODFEN bits. Disabling the SPI using the SPE bit will cause a partial reset of the SPI and may cause the loss of a message currently being received or transmitted.

To clear the MODF flag, write 1 to the MODF bit in the SPSCR. The clearing mechanism must occur with no MODF condition existing or else the flag is not cleared.

11.10 Module Memory Map

Four registers (**Table 11-3**) control and monitor SPI operations**.** Two of the four registers are read/write registers. They should be accessed only with word accesses. Accesses other than word lengths result in undefined results. Two registers are *read-* and *write-only.*

Table 11-3. SPI Module Memory Map (SPI_BASE = $1FFFE8)

Address Offset	Register Acronym	Register Name	Access Type	**Chapter Location**
Base + $0	SPSCR	SPI Status and Control Register	Read/Write	**Section 11.11.1**
Base + $1	SPDSCR	SPI Data Size and Control Reg.	Read/Write	**Section 11.11.2**
Base + $2	SPDRR	SPI Data Receive Register	*Read-Only*	**Section 11.11.3**
Base + $3	SPDTR	SPI Data Transmit Register	*Write-Only*	**Section 11.11.4**

Note: SPI registers are not available on the 56855 chip

Add. Offset	**Register Name**		**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
$0	SCSCR	R	SPR			DSO	ERRIE	MODF EN	SPRIE	SPMS TR	CPOL	CPHA	SPE	SPTIE	SPRF	OVRF	MODF	SPTE
		W																
$1	SPDSCR	R	WOM	TDMA EN	RDMA EN	0	0	0	0	0	0	0	0	0	TDS			
		W																
$2	SPDRR	R	R15	R14	R13	R12	R11	R10	R9	R8	R7	R6	R5	R4	R3	R2	R1	R0
		W																
$3	SPDTR	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		W	T15	T14	T13	T12	T11	T10	T9	T8	T7	T6	T5	T4	T3	T2	T1	T0

R	0	Read as 0
W		Reserved

Figure 11-13. SPI Register Map Summary

11.11 SPI Register Descriptions (SPI_BASE = $1FFFE8)

Table 11-3 lists the SPI registers in ascending address, including the acronym, bit names, and address of each register. These read/write registers should be accessed only with word accesses. Accesses other than word lengths result in undefined results.

11.11.1 SPI Status and Control Register (SPSCR)

The SPSCR register:

- Enables SPI module interrupt requests
- Selects interrupt requests
- Configures the SPI module as Master or Slave
- Selects serial clock polarity and phase
- Enables the SPI module Receive Data register full interrupt
- Enables Transmits Data register empty interrupt
- Selects Master SPI baud rate

Base + $0	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	SPR			DSO	ERRIE	MODFEN	SPRIE	SPMSTR	CPOL	CPHA	SPE	SPTIE	SPRF	OVRF	MODF	SPTE
Write																
Reset	0	1	1	0	0	0	0	1	0	1	0	0	0	0	0	0

Figure 11-14. SPI Status and Control Register (SPSCR)

See Programmer's Sheets on Appendix page B-64

Note: Using BFCLR or BFSET instructions to modify SPSCR can cause unintended side effects on the status bits.

11.11.1.1 SPI Baud Rate Select Bits (SPR)—Bits 15–13

While in the Master mode, these read/write bits select one of eight baud rates depicted in **Table 11-4.** SPR2:0 have no effect in Slave mode. Reset clears SPR2:0 to b011. Use the formula below to calculate the SPI baud rate.

SPR1 and SPR0 have no effect in Slave mode. Reset clears SPR1 and SPR0. Use the formula below to calculate the SPI baud rate.

$$\text{Baud Rate} = \frac{\text{CLK}}{\text{BD}}$$

CLK = Peripheral Bus Clock BD = Baud Rate Divisor

Table 11-4. SPI Master Baud Rate Selection

SPR[2:0]	Baud Rate Divisor (BD)
000	2
001	4
010	8
011	16
100	32
101	64
110	128
111	256

Note: The maximum data transmission rate for the SPI is typically limited by the bandwidth of the I/O drivers on the chip. Typical technology limits are normal at 40MHz and 10MHz for Wired OR. These apply to both Master and Slave modes. The BD field needs to be set to keep the module within these ranges.

11.11.1.2 Data Shift Order (DSO)—Bit 12

This read/write bit determines whether the MSB or LSB bit is transmitted or received first. Both Master and Slave SPI modules must transmit and receive the same length packets. Regardless how this bit is set, when reading from the SPDRR or writing to the SPDTR, the LSB will always be at bit location zero. If the data length is less than 16 bits, the data will be zero padded on the upper bits.

- 0 = MSB transmitted first (MSB > LSB)
- 1 = LSB transmitted first (LSB > MSB)

11.11.1.3 Error Interrupt Enable (ERRIE)—Bit 11

This read/write bit enables the MODF and OVRF bits to generate interrupt requests. Reset clears the ERRIE bit. The Error Interrupt Enable (ERRIE) bit enables both the MODF and OVRF bits to generate a receiver/error interrupt request.

- 0 = MODF and OVRF cannot generate interrupt requests
- 1 = MODF and OVRF can generate interrupt requests

11.11.1.4 Mode Fault Enable (MODFEN)—Bit 10

This read/write bit, when set to one, allows the MODF flag to be set. If the MODF flag is set, clearing the MODFEN does not clear the MODF flag.

If the MODFEN bit is low, the level of the $\overline{SS}$ pin does not affect the operation of an enabled SPI configured as a master. For an enabled SPI configured as a slave, having MODFEN low only prevents the MODF flag from being set. If configured as a Master and MODFEN = 1, a transmission in progress will stop if $\overline{SS}$ goes low. It does not affect any other part of SPI operation.

The Mode Fault Enable (MODFEN) bit can retard the MODF flag from being set. The retarded bit results in only the OVRF bit being enabled by the ERRIE bit. This enabling generates Receiver/Error Interrupt requests.

11.11.1.5 SPI Receiver Interrupt Enable (SPRIE)—Bit 9

This read/write bit enables interrupt requests generated by the SPRF bit. The SPRF bit is set when a full data length transfers from the Shift register to the SPI Data Receive Register (). The SPI Receiver Interrupt Enable (SPRIE) bit enables the SPRF bit to generate receiver interrupt requests regardless of the state of the SPE bit. The clearing mechanism for the SPRF flag is always just a read to the SPDRR.

- 0 = SPRF interrupt requests disabled
- 1 = SPRF interrupt requests enabled

11.11.1.6 SPI Master (SPMSTR)—Bit 8

This read/write bit selects master mode operation or slave mode operation.

- 0 = Slave mode
- 1 = Master mode (default)

11.11.1.7 Clock Polarity (CPOL)—Bit 7

This read/write bit determines the logic state of the SCLK pin between transmissions. To transmit data between SPI modules, the SPI modules must have identical CPOL values.

- 0 = Falling edge of SCLK starts transmission
- 1 = Rising edge of SCLK starts transmission

11.11.1.8 Clock Phase (CPHA)—Bit 6

This read/write bit controls the timing relationship between the serial clock and SPI data. To transmit data between SPI modules, the SPI modules must have identical CPHA values. When CPHA = 0, the $\overline{SS}$ pin of the slave SPI module must be set to Logic 1 between full length data transmissions. *Do not use CPHA = 0 while in the DMA mode.*

11.11.1.9 SPI Enable (SPE)—Bit 5

This read/write bit enables the SPI module. Clearing SPE causes a partial reset of the SPI. When setting/clearing this bit, *no* other bits in the SPSCR should be changed. Failure to following this statement may result in spurious clocks.

- 0 = SPI module disabled
- 1 = SPI module enabled.

11.11.1.10 SPI Transmit Interrupt Enable (SPTIE)—Bit 4

This read/write bit enables interrupt requests generated by the SPTE bit. SPTE is set when a full data length transfers from the SPI Data Transmit Register (SPDTR) to the Shift register. The SPI Transmitter Interrupt Enable (SPTIE) bit enables the SPTE flag to generate transmitter interrupt requests, provided the SPI is enabled (SPE = 1). The clearing mechanism for the SPTE flag is always just a write to the SPDTR.

- 0 = SPTE interrupt requests disabled
- 1= SPTE interrupt requests enabled

11.11.1.11 SPI Receiver Full (SPRF)—Bit 3

This *read-only* flag is set each time full length data transfers from the Shift register to the SPI Data Receive Register (SPDRR). SPRF generates an interrupt request if the SPRIE bit in the SPI Status and Control Register (SPSCR) is set also. This bit may not be cleared.

- 0 = Data Receive register not full
- 1 = Data Receive register full

11.11.1.12 Overflow (OVRF)—Bit 2

This *read-only* flag is set if software does not read the data in the SPI Data Receive Register (SPDRR) before the next full data enters the Shift register. In an overflow condition, the data already in the SPDRR is unaffected, and the data shifted in last is lost. Clear the OVRF bit by reading the SPI Status and Control Register (SPDSCR) with OVRF set and then reading the SPDRR. This bit may be cleared using the proper software sequence.

- 0 = No overflow
- 1 = Overflow

11.11.1.13 Mode Fault (MODF)—Bit 1

This *read-only* flag is set in a slave SPI if the $\overline{SS}$ pin goes high during a transmission with the MODFEN bit set. In a master SPI, the MODF flag is set if the $\overline{SS}$ pin goes low at any time with the MODFEN bit set. Clear the MODF bit by writing one to the MODF bit when it is set. The delayed bit results in only the OVRF interrupt being enabled by the ERRIE bit. This enabling generates Receiver/Error Interrupt requests.

- 0 = $\overline{SS}$ pin at appropriate logic level
- 1 = $\overline{SS}$ pin at inappropriate logic level

11.11.1.14 SPI Transmitter Empty (SPTE)—Bit 0

This *read-only* flag is set each time the SP Data Transmit Register (SPDTR) transfers a full data length into the Shift register. SPTE generates an interrupt request if the SPTIE bit in the SPI Status and Control Register (SPSCR) is set also. This bit may be cleared using the proper software sequence.

- 0 = Data Transmit register not empty
- 1 = Data Transmit register empty

Note: Do not write to the SPI Data Register unless the SPTE bit is high.

11.11.2 SPI Data Size and Control Register (SPDSCR)

This read/write register determines the data length for each transmission. The master and slave must transfer the same size data on each transmission. A new value will only take effect at the time the SPI is enabled (SPE bit in SPSCR register set from zero to one). In order to have a new value take effect, disable then re-enable the SPI with the new value in the register. The SPDSCR:

- Enables SPI DMA mode for the Transmitter and Receiver
- Enables Wired OR mode on MISO and MOSI
- Configures the size of the transmission

Base +$1	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	WOM	TDMAEN	RDMAEN	0	0	0	0	0	0	0	0	0	TDS			
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1

Figure 11-15. SPI Data Size and Control Register (SPDSCR)

See Programmer's Sheets on Appendix page B-65

11.11.2.1 Wired OR Mode (WOM)—Bit 15

This control bit is used to select the nature of the SPI pins. When enabled, the WOM bit is set, the SPI pins are configured as open-drain drivers with the pull-ups disabled. However, when disabled, the WOM bit is cleared, and the SPI pins are configured as push-pull drivers.

- 0 = Wired OR mode disabled
- 1 = Wired OR mode enabled

11.11.2.2 Transmitter DMA Enable (TDMAEN)—Bit 14

In the DMA mode, normal interrupts (not error) are suppressed and a DMA Request is generated instead. The requirement for clearing SPRF/SPTE changes to read/write the SPDRR/SPDTR, thereby eliminating the read of SPSCR.

- 0 = Transmitter DMA mode disabled
- 1 = Transmitter DMA mode enabled

11.11.2.3 Receiver DMA Enable (RDMAEN)—Bit 13

In the DMA mode, normal interrupts (not error) are suppressed and a DMA Request is generated instead. The requirement for clearing SPRF/SPTE changes to read/write the SPDRR/SPDTR, thereby eliminating the read of SPSCR.

- 0 = Receiver DMA mode disabled
- 1 = Receiver DMA mode enabled

11.11.2.4 Reserved—Bits 12–4

This bit field is reserved or not implemented. It is read as 0, but cannot be modified by writing.

11.11.2.5 Transmission Data Size (TDS)—Bits 3–0

Please see **Table 11-5** for detailed transmission data.

Table 11-5. Transmission Data Size

DS3 - DS0	Size of Transmission
0	Not Allowed
1	2 bits
2	3 bits
3	4 bits
4	5 bits
5	6 bits
6	7 bits
7	8 bits
8	9 bits
9	10 bits
10	11 bits
11	12 bits
12	13 bits
13	14 bits
14	15 bits
15	16 bits

11.11.3 SPI Data Receive Register (SPDRR)

This *read-only* register will show the last full data received after a complete transmission while the SPRF bit will set when new data has been transferred to this register.

Base + $2	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	R15	R14	R13	R12	R11	R10	R9	R8	R7	R6	R5	R4	R3	R2	R1	R0
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 11-16. SPI Data Receive Register (SPDRR)

See Programmer's Sheets on Appendix page B-66

11.11.3.1 Data Receive—Bits 15–0

11.11.4 SPI Data Transmit Register (SPDTR)

This *write-only* register modifies the data to the transmit data buffer. When the SPTE bit is set, new data should be written to this register. If new data is not written while in the Master mode, a new transaction will not be initiated until this register is written. When in Slave mode, the old data will be re-transmitted. All data should be written with the LSB at bit zero.

Base + $3	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Write	T15	T14	T13	T12	T11	T10	T9	T8	T7	T6	T5	T4	T3	T2	T1	T0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 11-17. SPI Data Transmit Register (SPDTR)

See Programmer's Sheets on Appendix page B-67

11.11.4.1 Data Transmit—Bits 15–0

11.12 Resets

Any system reset completely resets the SPI. Partial resets occur whenever the SPI enable bit (SPE) is low. Whenever SPE is low, the following will occur:

- The SPTE flag is set.
- Any slave mode transmission currently in progress is aborted.
- Any master mode transmission currently in progress is continued to completion.
- The SPI state counter is cleared, making it ready for a new complete transmission.
- All the SPI port logic is disabled.

The following items are reset only by a system reset:

- The SPDTR and SPDRR Registers
- All control bits in the SPSCR Register (MODFEN, ERRIE, SPR1, and SPR0)
- The status flags SPRF, OVRF, and MODF

By not resetting the control bits when SPE is low, it is possible to clear SPE between transmissions without having to set all control bits again when SPE is set back high for the next transmission.

By not resetting the SPRF, OVRF, and MODF flags, it is possible to service the interrupts after the SPI has been disabled. Disable SPI by writing 0 to the SPE bit. SPI can also be disabled by a Mode Fault occurring in a SPI configured as a Master with MODF.

11.13 Interrupts

Four SPI status flags can be enabled to generate interrupt requests.

Table 11-6. SPI Interrupts

Flag	Request
SPTE (Transmitter Empty)	SPI Transmitter Interrupt Request (SPTIE = 1,SPE = 1)
SPRF (Receiver Full)	SPI Receiver Interrupt Request (SPRIE = 1)
OVRF (Overflow)	SPI Receiver/Error Interrupt Request (ERRIE = 1)
MODF (Mode Fault)	SPI Receiver/Error Interrupt Request (ERRIE = 1)

The following sources in the SPI Status and Control Register can generate interrupt requests:

- The SPI Transmitter Interrupt Enable (SPTIE) bit enables the SPTE flag to generate transmitter interrupt requests provided the SPI is enabled (SPE = 1). The clearing mechanism for the SPTE flag is always just a write to the Data Transmit register.
- The SPI Receiver Interrupt Enable (SPRIE) bit enables the SPRF bit to generate receiver interrupt requests regardless of the state of the SPE bit. The clearing mechanism for the SPRF flag is always just a read to the Data Receive register.
- The Error Interrupt Enable (ERRIE) bit enables both the MODF and OVRF bits to generate a receiver/error interrupt request.
- The Mode Fault Enable (MODFEN) bit can prevent the MODF flag from being set so only the OVRF bit is enabled by the ERRIE bit to generate receiver/error interrupt requests.

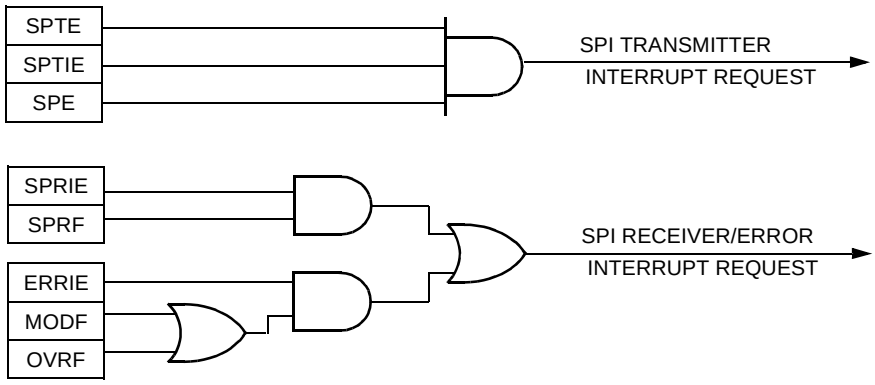


Figure 11-18. SPI Interrupt Request Generation

The following sources in the SPI Status and Control Register can generate interrupt requests:

- SPI Receiver Full (SPRF) bit — The SPRF bit becomes set every time a full data transmission transfers from the Shift register to the receive data register. If the SPI Receiver Interrupt Enable (SPRIE) bit is also set, SPRF can generate a SPI receiver/error interrupt request.
- SPI Transmitter Empty (SPTE) — The SPTE bit becomes set every time a full data transmission transfers from the Data Transmit register to the Shift register. If the SPI Transmit Interrupt Enable (SPTIE) bit is also set, SPTE can generate a SPTE interrupt request.

Chapter 12
Enhanced Synchronous Serial Interface (ESSI)

12.1 Introduction

This chapter describes the Enhanced Synchronous Serial Interface (ESSI), discusses the architecture, the programming model, the operating modes, and the initialization of the ESSI. The ESSI is a full-duplex, serial port designed to allow Digital Signal Controllers (DSCs) to communicate with a variety of serial devices, including industry-standard codecs, other DSCs, and microprocessors. It is typically used to transfer samples in a periodic manner. The ESSI consists of independent transmitter and receiver sections with independent clock generation and frame synchronization.

12.2 Features

ESSI capabilities include:

- Independent (asynchronous) or shared (synchronous) transmit and receive sections with separate or shared internal/external clocks and frame syncs
- Normal mode operation using frame sync
- Network mode operation allowing multiple devices to share the port with as many as 32 time slots
- Network mode enhancements
  - time slot mask registers (receive and transmit)
  - end of frame interrupt
- Programmable internal clock divider
- Programmable word length (8, 10, 12, or 16 bits)
- Program options for frame sync and clock generation
- ESSI power-down feature
- Audio enhancements
  - three transmitters per ESSI (six-channel surround sound possible, but requires two ESSIs)

12.3 Signal Descriptions

12.3.1 Signal Properties

Table 12-1. Signal Properties

Name	Type	Function	Reset State	Notes
SCK	I/O	ESSI Transmit Clock	Input	Controlled by Reset State of SCKD bit in SCR4 register
SC2	I/O	ESSI Transmit Frame Sync	Input	Controlled by Reset State of SCD2 bit in SCR4 register
SC0	I/O	ESSI Receive Clock	Input	Controlled by Reset State of SCD0 bit in STR4 register
SC1	I/O	ESSI Receive Frame Sync	Input	Controlled by Reset State of SCD1 bit in STR4 register
STD	Output	ESSI Transmit Data	Hi-Z	Since ESSIEN bit is SCR2 register is reset to 0
SRD	Input	ESSI Receive Data	—	—

12.3.2 External Signals Descriptions

Three to six signals are required for ESSI operation, depending on the selected operation mode. The function of the I/O pins for the ESSI is determined by a number of control bits. **Table 12-2** lists the controls affecting this and the resulting pin definitions. The Serial Transmit Data (STD) signal and Serial Control (SC0 and SC1) signals are fully synchronized to the clock if they are programmed as transmit data signals. **Table 12-3** further clarifies the possible configuration of the SC0 and SC1 pins during synchronous modes when they are not being used as transmit data pins.

In addition to the functional configuration of the I/O pins, the direction of clocks and frame syncs can be configured. The source of the transmit and receive clocks used by ESSI is determined by three control bits, illustrated in **Table 12-1.** The source of the receive and transmit frame syncs is determined by three control bits, illustrated in **Table 12-3.**

Table 12-2. Mode and Signal Definition Table

Control Bits					ESSI Signal Definitions					
SYN	TE0	TE1	TE2	RE	SC0	SC1	SC2	SCK	STD	SRD
0	0	X	X	0	U					
				1	RXC	FSR	U			RD
	1			0	U		FST	TXC	TD0	U
				1	RXC	FSR				RD
1	0	0	0	0	U					
				1	F0/U	F1/T0D/U	FS	XC	U	RD
			1	0		TD2				U
				1						RD
		1	0	0	TD1	F1/T0D/U				U
				1						RD
			1	0		TD2				U
				1						RD
	1	0	0	0	F0/U	F1/T0D/U			TD0	U
				1						RD
			1	0		TD2				U
				1						RD
		1	0	0	TD1	F1/T0D/U				U
				1						RD
			1	0		TD2				U
				1						RD

F0=Flag 0

F1 =Flag 1 if SSC1 = 0

FS =Transmit / Receive Frame Sync (Synchronous Operation)

FSR = Receiver Frame Sync

FST= Transmitter Frame Sync

RD = Receiver Data

RXC =Receiver Clock

T0D =Transmitter 0 Drive Enable if SSC1=1 & SCD1=1

TD0 =Transmit Data Signal 0

TD1 =Transmit Data Signal 1

TD2 =Transmit Data Signal 2

TXC =Transmitter Clock

X =Indeterminate

XC =Transmitter / Receiver Clock (Synchronous Operation)

U =Unused (May Be Used As GPIO Signal)

Table 12-3. ESSI Clock Sources

Controls			RX Clock		TX Clock	
SYN	SCKD	SCD0	Pin	Source	Pin	Direction
Asynchronous						
0	0	0	SC0	External	SCK	External
0	0	1	SC0	Internal	SCK	External
0	1	0	SC0	External	SCK	Internal
0	1	1	SC0	Internal	SCK	Internal
Synchronous						
1	0	x	SCK	External	SCK	External
1	1	x	SCK	Internal	SCK	Internal

Table 12-4. ESSI Frame Sync Sources

Controls			RX Clock Frame Sync		TX Frame Sync	
SYN	SCD2	SCD1	Pin	Source	Pin	Direction
Asynchronous						
0	0	0	SC1	External	SC2	External
0	0	1	SC1	Internal	SC2	External
0	1	0	SC1	External	SC2	Internal
0	1	1	SC1	Internal	SC2	Internal
Synchronous						
1	0	x	SC2	External	SC2	External
1	1	x	SC2	Internal	SC2	Internal

12.3.2.1 ESSI Transmit Clock (SCK)

This pin can be configured as either an input or an output pin. This clock signal is used by all the enabled transmitters and the receiver in Synchronous modes or by Transmitter 0 in Asynchronous modes. Please see **Table 12-3** for details about how to configure this pin.

Note: Although an external serial clock can be independent of and asynchronous to the system clock, the external ESSI clock frequency must not exceed $F_{IPBus\_CLK}/4$, and each ESSI clock phase must exceed the minimum of 1.5 IPBus_CLK cycles.

12.3.2.2 ESSI Serial Control Signal 0 (SC0)

The function of this signal is determined by the programmer selecting either Synchronous or Asynchronous mode, according to data in **Table 12-4.** In Asynchronous mode, this signal is used for the receive clock I/O. In Synchronous mode, this signal is used as the transmitter data out

signal for Transmit Shift register TX1 or for Serial Flag I/O. A typical application of Serial Flag I/O would be multiple device selection for addressing in codec systems.

If SC0 is configured as a serial flag signal or receive clock signal, its direction is determined by the Serial Control Direction 0 (SCD0) bit in the ESSI Control Register 4 (SCR4). When configured as an output, SC0 may be used either as the serial Output Flag 0 (OF0) or as a Receive Shift register clock output. If the SC0 is used as the serial OF0, its value is determined by the value of the serial OF0 bit in the SCR4 register.

If SC0 is an input, this signal may be used either as serial Input Flag 0 (IF0) or as a Receive Shift Register (RSR) clock input. If SC0 is used as serial Input Flag 0, it controls the state of the serial Input Flag 0 bit in the ESSI Status Register (ESSR). When SC0 is configured as a transmit data signal, it is always an output signal regardless of the SCD0 bit value. SC0 is fully synchronized with the other transmit data signals, STD and SC1.

Note: The ESSI can operate with more than one active transmitter only in the Synchronous mode.

12.3.2.3 ESSI Receive Frame Sync (SC1)

The function of this signal is determined by the programmer selecting either Synchronous or Asynchronous modes, according to **Table 12-4.** In the Asynchronous mode, such as a single codec with asynchronous transmit and receive, the SC1is the receiver frame sync I/O. In the Synchronous mode, the SC1 is used for the transmitter data out signal of Transmit Shift Register (TXSR2), for the transmitter 0 drive-enable signal, or for serial flag I/O.

When used as serial flag I/O, SC1operates like SC0. SC0 and SC1 are independent flags, but they may be used together for multiple serial device selection. SC0 and SC1 can be used without being encoded to select up to two codecs. They may also be decoded externally to select up to four codecs. If the SC1 is configured as a serial flag signal, its direction is determined by the SCD1 bit in SCR4 register.

If the SC1 is configured as a serial flag or receive frame sync signal, its direction is determined by the Serial Control Direction 1 (SCD1) bit in the SCR4 register.

When configured as an output, the SC1 signal can be used as a serial output flag, as the transmitter 0 drive-enable signal for the purpose to control an external high-drive buffer, or as the receive frame sync signal output. If SC1 is used as serial output flag 1, its value is determined by the value of the serial Output Flag 1 (OF1) bit in the SCR4 register.

When configured as an input, this signal can be used to receive frame sync signals from an external source, or it can be used as a serial input flag. When SC1 is a serial input flag, it controls status bit IF1 in the SSR.

When this signal is configured as a transmit data signal, it is always an output signal regardless of the SCD1 bit value. As an output, it is fully synchronized with the other ESSI transmit data signals (STD and SC0).

Note: The source of the Transmitter 0 drive-enable signal is the STD_$\overline{\text{OEN}}$ signal.

12.3.2.4 ESSI Serial Control Signal 2 (SC2)

This pin is used for frame sync I/O and can be configured as either an input or an output pin. The direction of this signal is determined by the SCD2 bit of the SCR4 register. The frame sync is used by the transmitter to synchronize the transfer of data in Asynchronous mode. It is used by both the receiver and transmitter in Synchronous mode. The frame sync signal can be one-bit, or one-word in length. The start of the frame sync can occur one-bit before the transfer of data, or right at the start of the data transfer.

12.3.2.5 ESSI Transmit Data (STD)

The STD signal is used for transmitting data from the TXSR0 serial Transmit Shift Register. STD is an output pin when data is being transmitted from the TXSR0 Shift Register. The STD pin is tri-stated after transmitting the last data bit when another data word does not follow immediately. If a data word follows immediately within a full clock cycle, the STD pin is not tri-stated.

12.3.2.6 ESSI Receive Data (SRD)

The SRD signal receives serial data, then transfers the data to the ESSI Receive Data Shift Register (RXSR).

12.4 ESSI Block Diagram

The following figure illustrates the ESSI block diagram. It consists of five control registers to set-up the port, one status register, separate transmit and receive circuits with FIFO registers, and separate serial clock and frame sync generation for the transmit and receive sections.

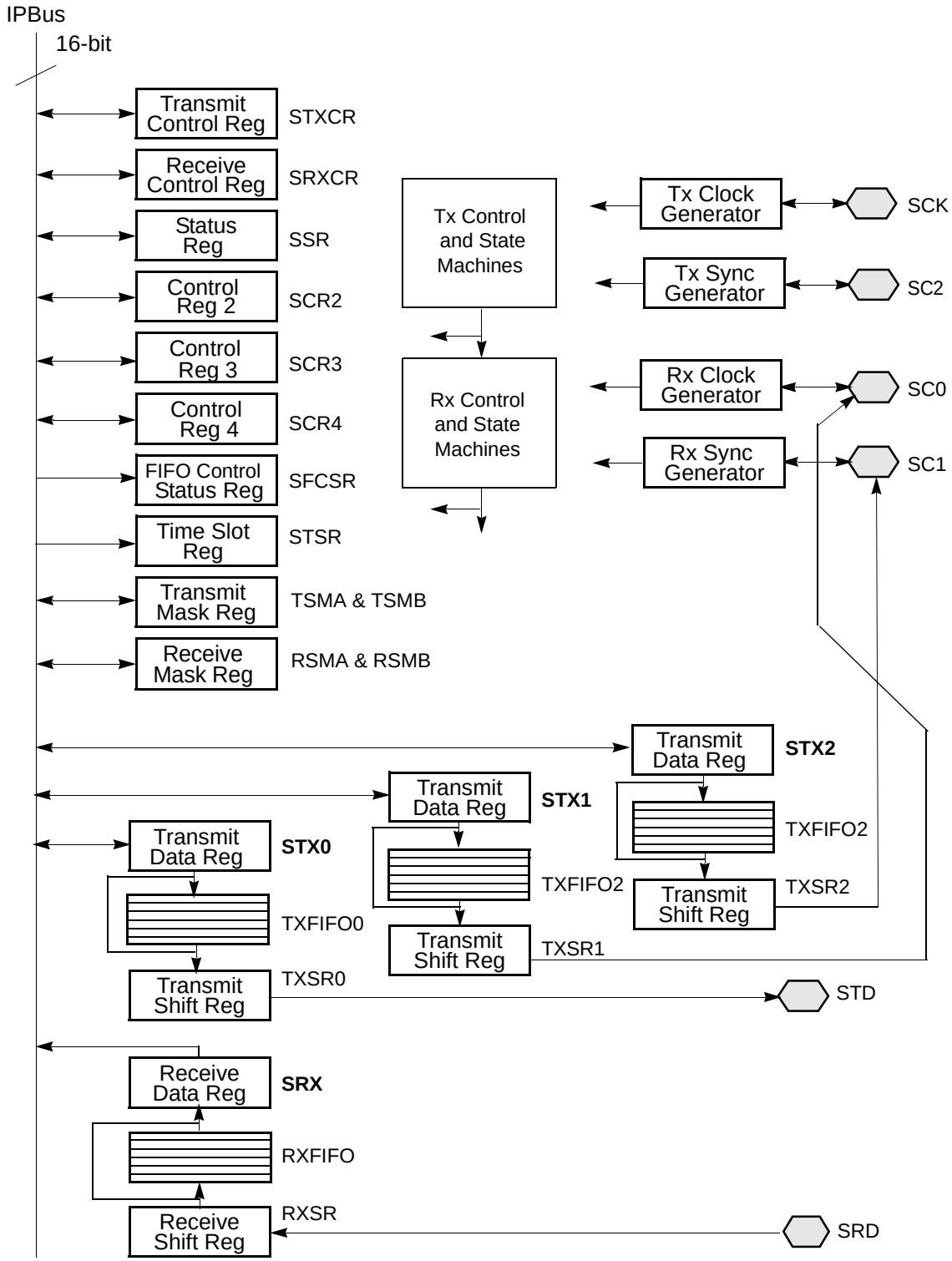


Figure 12-1. ESSI Block Diagram

12.5 Functional Description

The ESSI has two basic operating modes. **Table 12-5** lists these operating modes and some of the typical applications. These distinctions result in the basic operating modes allowing the ESSI to communicate with a wide variety of devices. These modes can be programmed by several bits in the ESSI control registers. For additional information, please see **Section 12.8.**

Table 12-5. ESSI Operating Modes

TX, RX Sections[1]	Serial Clock	Mode[2]	Typical Application
Asynchronous	Continuous	Normal	Multiple Asynchronous Codecs
Asynchronous	Continuous	Network	TDM Codec or DSC Networks
Synchronous	Continuous	Normal	Multiple Synchronous Codecs
Synchronous	Continuous	Network	TDM Codec or DSC Network

1. In the Synchronous mode, the transmitter and receiver use a common clock and frame sychronization signal. In the Asynchronous mode the transmitter and receiver operate independently on their own clocks and frame syncs.
2. In the Normal mode, the ESSI only transmits during the first time-slot of each I/O frame. In the Network mode, any number from 2 to 32 data words of I/O per frame can be used. The Network mode is typically used in star or ring time division multiplex networks with other processors or codecs, allowing interface to TDM networks without additional logic.

The ESSI supports both Normal and Network modes, and these can be selected independently of whether the transmitter and receiver are synchronous or asynchronous. Typically these protocols are used in a periodic manner, where data is transferred at regular intervals, such as at the sampling rate of an external codec. Both modes use the concept of a frame. The beginning of the frame is marked with a frame sync.

The frame sync occurs at a periodic interval. The length of the frame is determined by the DC and WL field bits in either the SRXCR or STXCR register, depending on whether data is being transmitted or received. The number of words transferred per frame depends on the mode of the ESSI.

12.5.1 Normal Mode

Normal mode is the simplest mode of the ESSI. It is used to transfer one word per frame. A frame sync occurs at the beginning of each frame. The length of the frame is determined by the following factors:

- The period of the serial bit clock (PSR, PM bits for internal clock or the frequency of the external clock on the SCK or SC0 pin)
- The number of bits per sample (WL) bits
- The number of time slots per frame (DC) bits

If the Normal mode is configured to provide more than one time slot per frame, data is transmitted only in the first time slot. No data is transmitted in subsequent time slots. **Figure 12-2** and **Figure 12-3** show sample timing of Normal mode transfers.

12.5.1.1 Normal Mode Transmit

Conditions for data transmission from the ESSI in the Normal mode:

1. Set the STXCR, SCR2, SCR3, and SCR4 registers to select Normal mode operation.
2. Define the transmit clock, transmit frame sync and frame structure required for proper system operation.
3. ESSI enabled (ESSIEN = 1)
4. Enable TXFIFO (TFEN = 1) and configure the Transmit Watermark (TFWM = n) if this TXFIFO is used.
5. Write data to the Transmit Data (STX) register.
6. Enable transmit interrupts.
7. Set the TE bit (TE = 1) to enable the transmitter on the next frame sync boundary.

Figure 12-2 and **Table 12-6** describe the functions performed during transmit operation in this mode.

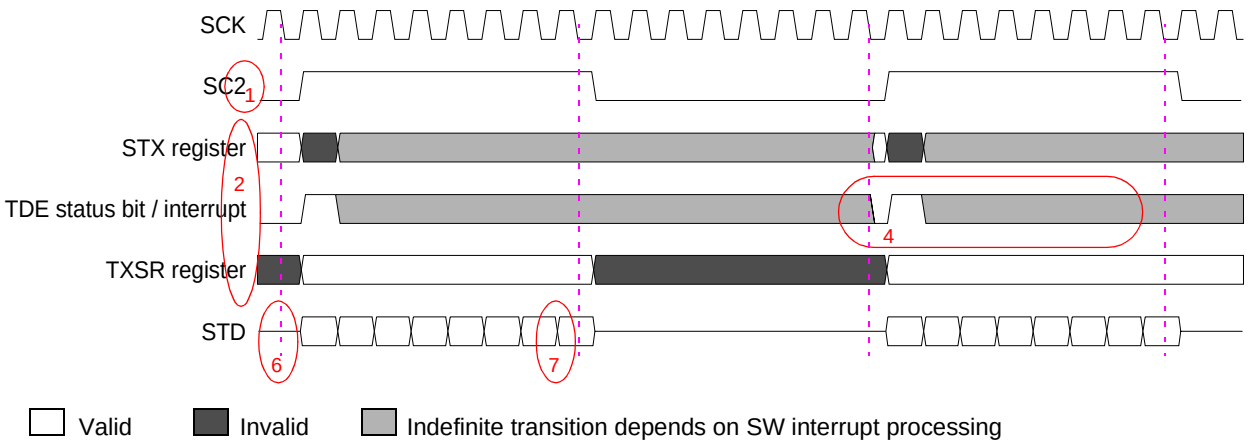


Figure 12-2. Normal Mode Transmit Timing (WL=8 Bit Words, DC = 1)

Table 12-6. Normal Mode Transmit Operations

Step		TXFIFO Disabled[1]	TXFIFO Enabled
1	Rising edge of SC2. Note a word length frame sync is shown. This only works if DC>0.	—	—
2	Data transferred to TXSR.	From STX	From TXFIFO
3	STD output pin is enabled[2] and the first bit of the TXSR register appears on the output.	—	—
4	Flag status update.	The TDE bit is set	The TFE bit is set if the level of data in the TXFIFO falls below the watermark level.
5	If the TIE bit is set, enabling transmit interrupts, then: Open options for processing the data transfer is either polling or DMA transfers.	Transmit interrupt occurs when TDE is set.	Transmit interrupt occurs when TFE set.
6	The TXSR is shifted on the next rising edge of the SCK and the next bit appears on the STD pin.	—	—
7	When WL bits (Section 11.8.14) have been sent the STD is tri-stated.	—	—
8	Transmit underrun (setting the TUE bit of the SSR register) is prevented by[3]:	New data is written to the STX before the TXSR tries to obtain new transmit data at the next frame sync.	New data is written to the STX before the TXSR tries to obtain data from an empty TXFIFO (this can be several frame times).
9	Repeat at step 1 on the next frame sync[4].	—	—

1. See **Figure 12-2.**
2. The STD output signal is disabled except during the data transmission period.
3. **Section 12.8.9** describes what happens when the TUE bit is set.
4. The frame sync must not occur earlier than it is configured in the STXCR as documented in **Section 12.8.11.**

12.5.1.2 Normal Mode Receive

The conditions for data reception from the ESSI are as follows:

1. Set the SRXCR, SCR2, SCR3, and SCR4 registers to select Normal mode operation.
2. Define the receive clock.
3. Receive frame sync and frame structure required for proper system operation.
4. ESSI enabled (ESSIEN = 1)
5. Enable RXFIFO (RFEN=1) and configure Receive Watermark (RFWM = n) if RXFIFO is used.
6. Enable receive interrupts.
7. Set the RE bit (RE = 1) to enable the receiver operation on the next frame sync boundary.

Figure 12-3 and **Table 12-7** describes the functions performed during receive operation in this mode.

Figure 12-3. Normal Mode Receive Timing (WL=8 bit words, DC=1)

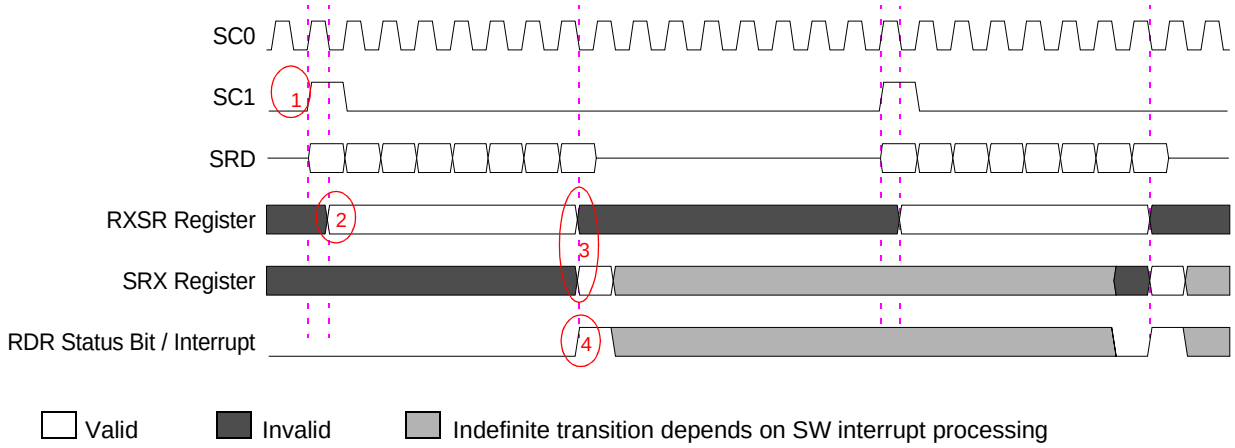


Table 12-7. Normal Mode Receive Operations

Step		TXFIFO Disabled[1]	TXFIFO Enabled
1	Leading edge of frame sync occurs on the SC1 pin.	—	—
2	Falling edge of receive clock occurs on the SC0 pin and the next bit of data is shifted into the RXSR register.	—	—
3	When WL bits[2] have been received. RXSR contents are transferred to the SRX register on the next falling edge of the receive clock. The SRX register is actually loaded during the middle of the last receive bit.	—	—
4	Flag status update	The RDR bit is set	The RFF bit is set if the level of data in the RSFIFO rises above the watermark level.
5	If the RIE bit is set, enabling receive interrupts, then: Other options for processing the data transfer is either polling or DMA transfers.	—	Receive interrupt occurs when RFF is set.
6	Receive overrun (setting the ROE bit of the SSR register) is prevented by[3]	—	Data is read from the SRX before the RXSR tries to provide more data to a full RSFIFO. It can take several frame times to fill the RXFIFO.
7	Repeat at set 1 on the next frame sync.[4]	—	—

1. See **Figure 12-2.**
2. See **Section 12.8.11.**
3. **Section 12.8.11** describes what happens when the ROE bit is set.
4. The frame sync must not occur earlier than it is configured in the STXCR as documented in **Section 12.8.11.**

12.5.2 Network Mode

Network mode is used for creating a Time Division Multiplexed (TDM) network, such as a TDM codec network or a network of DSCs. This mode only operates with continuous clock mode. A frame sync occurs at the beginning of each frame. In this mode, the frame is divided into more than one time slot. During each time slot, one data word can be transferred. Each time slot is then assigned to an appropriate codec or DSC on the network. The DSC can be a master device controlling its own private network, or a slave device connected to an existing TDM network, occupying a few time slots.

The frame sync signal indicates the beginning of a new data frame. Each data frame is divided into time slots and transmission and/or reception of one data word can occur in each time slot (rather than in just the frame sync time slot as in the Normal mode). The frame rate dividers, controlled by the DC bit field, select two to thirty-two time slots per frame. The length of the frame is determined by these factors:

- The period of the serial bit clock (PSR, PM[7:0] bits for internal clock, or the frequency of the external clock on the SCK and/or SC0 pins)
- The number of bits per sample (WL) bits
- The number of time slots per frame (DC) bit fields

Data can be transmitted in any time slot while in the Network mode. The distinction of the Network mode is each time slot is identified with respect to the frame sync (data word time). This time slot identification allows the option of transmitting data during the time slot by writing to the STX register or ignoring the time slot by writing to STSR. The receiver is treated in the same manner, except data is always being shifted into the RXSR and transferred to the SRX register. The core reads the SRX register, either using the data or discarding it.

Figure 12-4 and **Figure 12-5** provide sample timing of Network mode transfers. The figures illustrate Receive and Transmit frames of five time-slots for each. The numbered circles and arrows in the figure identify discussion notes contained in **Table 12-8** and **Table 12-9.**

12.5.2.1 Network Mode Transmit

The transmit portion of the ESSI is enabled when the ESSIEN and the TE bits in the SCR2 are both set. However, when the TE bit is set, the transmitter is enabled only after detection of a new frame boundary. Software has to find the start of the next frame by checking the TFS bit of the SSR. A normal start-up sequence for transmission is to do the following:

1. Set the STXCR, SCR2, SCR3, and SCR4 registers to select the Network mode operation.
2. Define the transmit clock.
3. Transmit frame sync and frame structure required for proper system operation.

4. ESSI enabled (ESSIEN = 1)
5. Enable TXFIFO (TFEN = 1) and configure the Transmit Watermark (TFWM = n) if this TXFIFO is used.
6. Write data to transmit data register (STX).
7. Enable transmit interrupts.
8. Set the TE bit (TE = 1) to enable the transmitter on the next frame sync boundary.

The transmitter timing for an 8-bit word with continuous clock, FIFO disabled, five words per frame sync, in the Network mode is shown in **Figure 12-4.** The explanatory notes for the transmit portion of the figure are provided in **Table 12-8.**

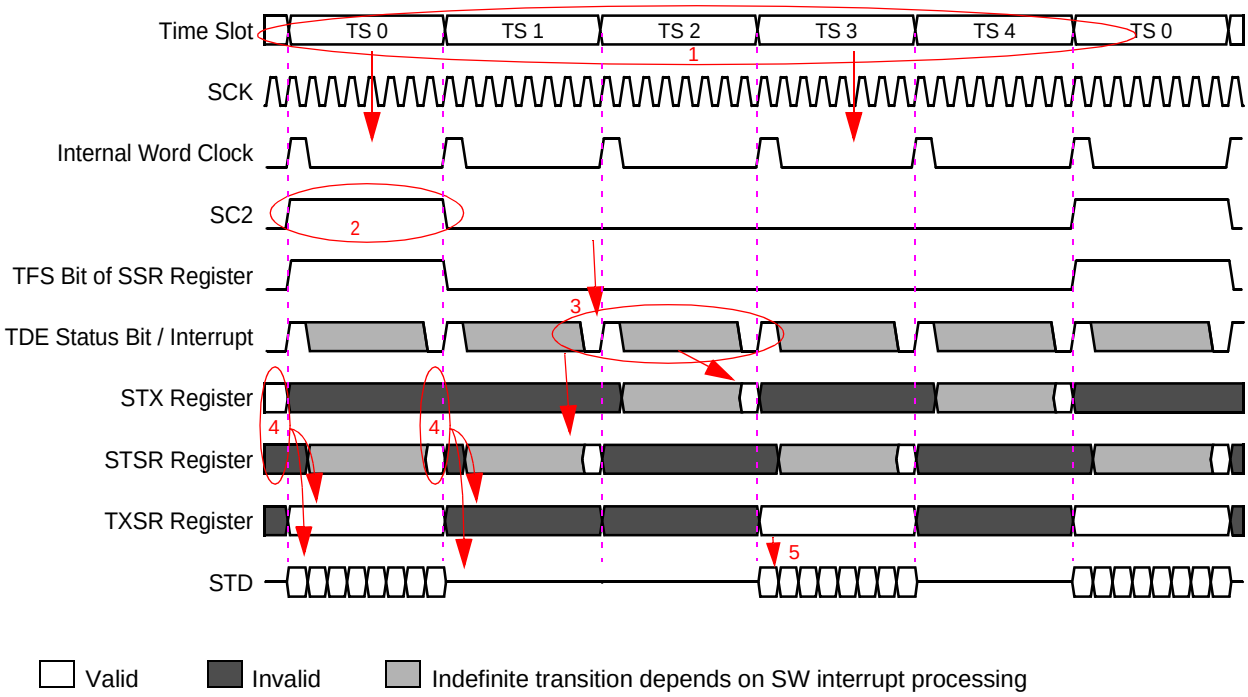


Figure 12-4. Network Mode Transmit Timing

Table 12-8. Notes for Transmit Timing in Figure 12-5

Note	Source Signal	Destination Signal	Description
1	—	—	Example of a five time-slot frame, transmitting in time-slots 0 and 3.
2	SC2	—	Example with word-length frame sync and standard timing (TFSI=0, TFSL=0, and TEFS=0). Frame timing begins with the rising edge of SC2.
3	—	TDE Status Flag and Interrupt	This flag is set at the beginning of each word to indicate that another data word should be supplied by the software. When the transmit interrupt is enabled, the processor is interrupted to request the data. The flag and interrupt are cleared when data is written to either the STX or STSR registers.[1]
4	STX / STSR Register	TXSR Register	On each word clock boundary a decision is made concerning what to transmit on the next time-slot. If the STSR register was written during the previous time-slot the STD pin is tri-stated. If the STSR register was NOT written during the previous time-slot the contents of the STX register is transferred to the TXSR register and this data is shifted out. If the STX register has not been written in the previous time-slot the previous data is reused. If neither of these registers were written in the previous time-slot the TUE status bit will be set and the hardware will operate as if the STX register had been written. The STD pin will be enabled and the contents of the STX will be transmitted again. This may lead to drive conflicts on the transmit data line.
5	TXSR Register	STD Pin	On active time-slots, the TXSR register contents are shifted out on the STD pin, one bit per rising edge of SCK. On inactive time-slots, the STD pin is tri-stated so it can be driven by another device.

1. **Section 12.13** provides a complete description of interrupt processing.

The operation of clearing the TE bit disables the transmitter after completion of transmission of the current data word. Setting the TE bit again enables transmission of the next word. During the time TE = 0, the STD signal is tri-stated. The TE bit should be cleared after the TDE bit is set, ensuring all pending data is transmitted.

Note: In case of Normal mode with external frame sync, TE bit should be cleared after the first bit of transmission only.

The Network mode transmitter generates interrupts every time slot (unless the TSM registers are utilized) and requires the DSC program to respond to each time slot. These responses may be one of the following:

- Write the Data register with data to enable transmission in the next time slot.
- Write the Time Slot register to disable transmission in the next time slot.
- Do nothing—transmit underrun occurs at the beginning of the next time slot and the previous data is re-transmitted.

12.5.2.2 Network Mode Receive

The receiver portion of the ESSI is enabled when both the ESSIEN and the RE bits in the SCR2 are set. However, when the RE bit is set, the receiver is enabled only after detection of a new frame boundary. Software has to find the start of the next frame by checking the RFS bit in the SSR. A normal start-up sequence for Receive operation is to do the following:

1. Set the SRXCR, SCR2, SCR3, and SCR4 registers to select Network mode operation, define the receive clock, receive frame sync and frame structure required for proper system operation.
2. ESSI enabled (ESSIEN = 1)
3. Enable RXFIFO (RFEN=1) and configure Receive Watermark (RFWM = n) if RXFIFO is used.
4. Enable receive interrupts.
5. Set the RE bit (RE = 1) to enable the receiver operation on the next frame sync boundary.

The receiver timing for an 8-bit word with continuous clock, FIFO disabled, five words per frame sync, in Network mode is shown in **Figure 12-5.** The explanatory notes for the receive portion of the figure are shown in **Table 12-9.**

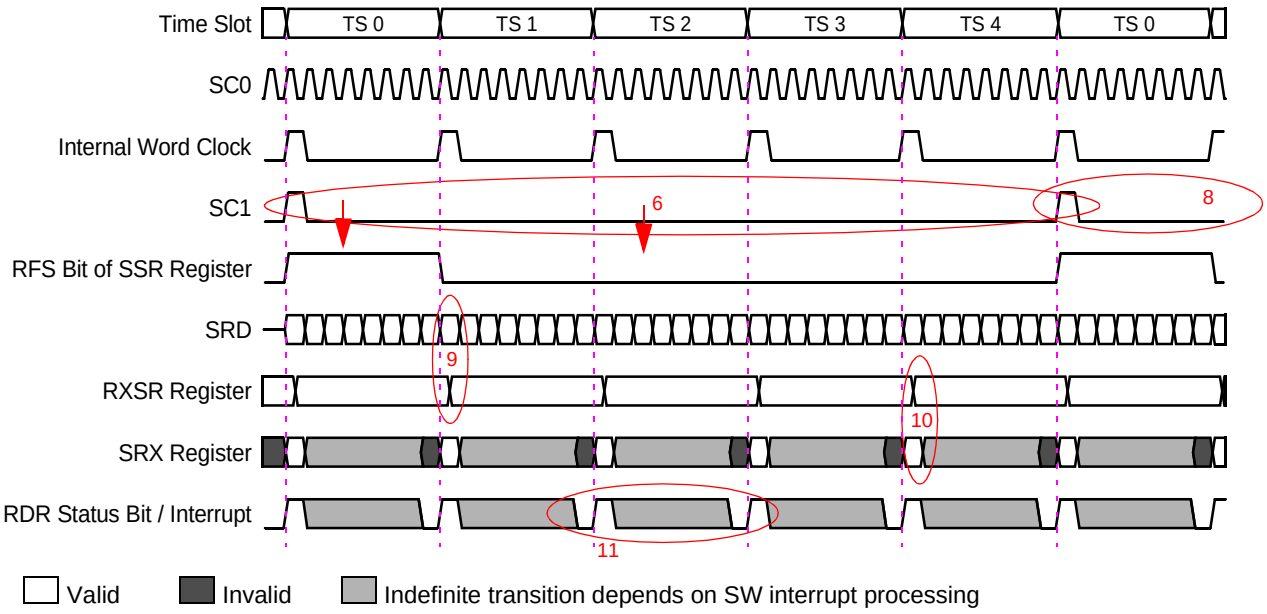


Figure 12-5. Network Mode Receive Timing

Table 12-9. Notes for Receive Timing in Figure 12-6

Note	Source Signal	Destination Signal	Description
6	—	—	Example of a five time slot frame, receiving data from time slots 0 and 2. The receive hardware will obtain data on the SRD pin every bit time. The software must determine which data belongs to each time slot and discard the unwanted time slot data.
7	SC0	—	Receive clock timing from which it is derived.
8	SC1	—	Example with bit length frame sync and standard timing (RFSI=0, RFSL=1, and REFS=0). Frame timing begins with the rising edge of SC1.
9	SRD	RXSR Register	Data on the SRD pin is sampled on the falling edge of SC0 and shifted into the RXSR register.
10	RXSR Register	SRX Register	At the word clock, the data in the RXSR register is transferred to the SRX register.
11	RDR Status Flag and Receive Interrupt	—	This flag is set for each word clock (time slot) indicating data is available to be processed. The software must keep track of the time slots as they occur so it knows which data to keep. If the receive interrupts are enabled (RIE=1) an interrupt will be generated when this status flag is set. The software reads the SRX register to clear the interrupt.[1]

1. **Section 12.13** provides a complete description of the interrupt process.

An interrupt can occur after the reception of each data word or the programmer can poll the RDR flag. The ESSI program response can be one of the following:

- Read SRX and use the data
- Read SRX and ignore the data
- Do nothing—the receiver overrun exception occurs at the end of the current time slot.

12.5.3 Synchronous/Asynchronous Operating Modes

The transmit and receive sections of the ESSI may be synchronous or asynchronous. During asynchronous operation the transmitter and receiver have their own separate clock and sync signals. When operating in the Synchronous mode the transmitter and receiver use common clock and synchronization signals, specified by the transmitter configuration. The SYN bit in SCR2 selects synchronous or asynchronous operation.

Since the ESSI is designed to operate either synchronously or asynchronously, separate receive and transmit interrupts are provided. During the synchronous operation, the receiver and transmitter operate in lock step with each other. Overhead may be reduced by eliminating either the receive or transmit interrupts, driving both channels from the same set of interrupts. If this decision is made, it is necessary to be aware of the specific timing of the receive and transmit interrupts, because the interrupts are not generated at the same exact point in the frame timing,

shown in **Figure 12-6.** If it is desired to run off a single set of interrupts, the TX interrupts should be used. If RX interrupts are used, there may be timing problems with the transmit data because this interrupt occurs a half-bit time before the transmit data is used by the hardware.

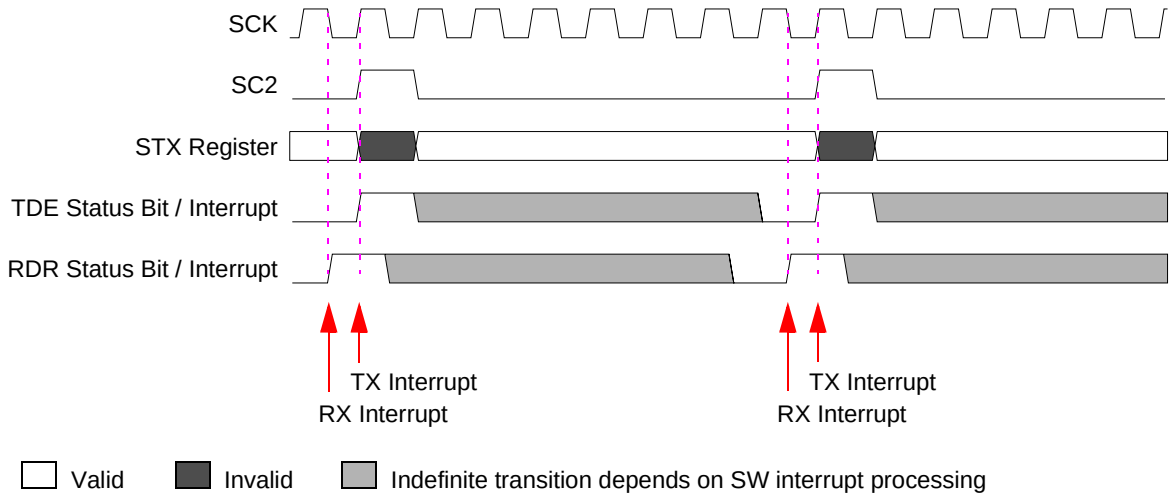


Figure 12-6. Synchronous Mode Interrupt Timing

12.5.4 Network Mode with Mask Registers Implemented

In order to reduce interrupt overhead, a number of enhancements were made to the ESSI module. The enhancements incorporate the mask registers (TSM and RSM).

Together, **Figure 12-8** and **Figure 12-33** illustrate sample timing of Network mode transfers using the mask registers. These figures duplicate the frame structure of **Figure 12-5** to illustrate the reduced amount of interrupt processing required. The numbered circles and arrows in the figures identify discussion notes contained in **Table 12-10** and **Table 12-11.**

12.5.4.1 Operation Using TSM Register

When the TSM register is included in the design interrupt overhead can be reduced. If all bits of the TSM register are set, the ESSI transmitter will continue to operate as previously described. The TSM register is used to disable the STD pin on specific time-slots. This is accomplished by writing the TSM with 0 in the time-slot bit location. Disabling a time- slot in this manner causes the time-slot to be ignored by the ESSI. This means no data is transferred to the Transmit Shift Register (TXSR), therefore interrupts are not generated for this time-slot. The transmitter timing, using TSM registers for an 8-bit word, continuous clock with disabled FIFO five words per frame sync in the Network mode is illustrated in **Figure 12-7.** Explanatory notes for the transmit portion of the figure are provided in **Table 12-10.**

Note: In this example there are only two transmit interrupts per frame instead of five as in the previous example where the TSM register is not used.

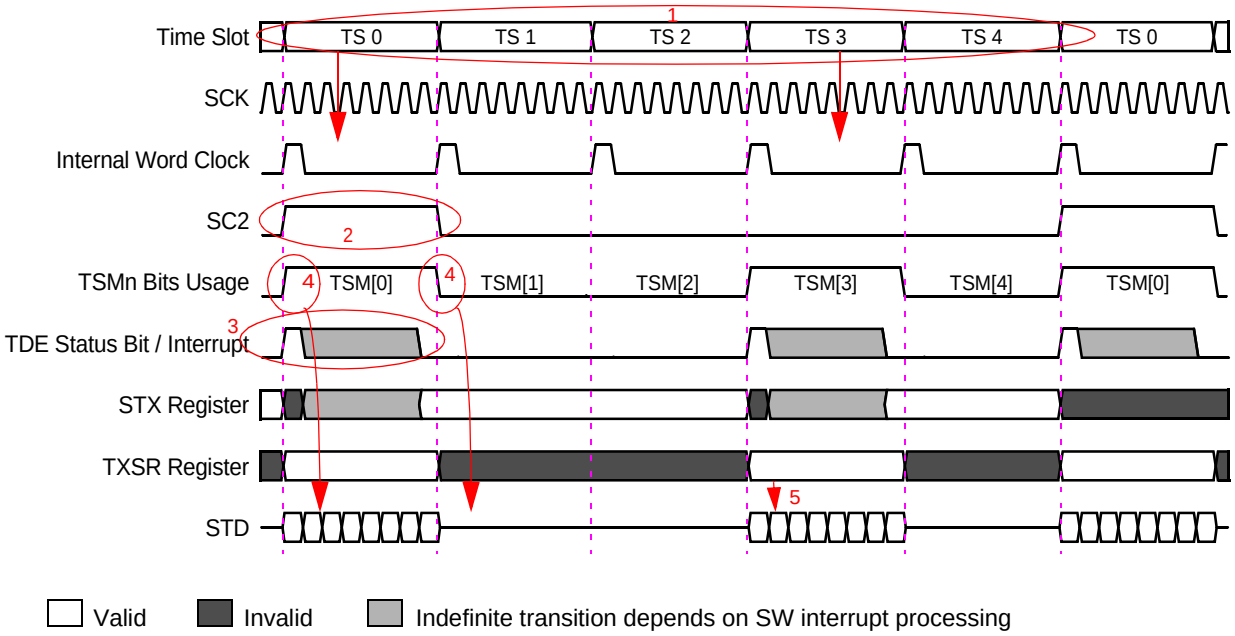


Figure 12-7. Network Mode Transmit Timing with Mask Register

Table 12-10. Notes for Transmit Timing with Mask Register in Figure 12-8

Note	Source Signal	Destination Signal	Description
1	—	—	Example of a five time slot frame, transmitting in time slots 0 and 3.
2	SC2	—	Example of a word length frame sync and standard timing (TFSI=0, TFSL=0, and TEFS=0). Frame timing begins with the rising edge of SC2.
3	—	TDE Status Flag and Interrupt	For enabled time slots, this flag is set at the beginning of each word to indicate the STX data has been used and another data word should be supplied by the software. If the transmit interrupt is enabled the processor is interrupted to request the data. The flag and interrupt are cleared when data is written to either the STX or STSR registers.[1]
4	STX / STSR Register	TXSR Register	On each word clock boundary a decision is made concerning what to transmit on the next time slot. If the TSMn register bit is **zero** for the next time slot, the STD pin is tri-stated and the time slot is ignored. When the TSMn bit is **one** for the next time slot, the contents or the STX register are transferred to the TXSR register and this data is shifted out. If the STX register has not been written in the previous time slot, the previous data is reused. **Note:** If the STSR is written instead of the STX, the STD pin is tri-stated as stated in **Section 12.5.2.1.** When neither of these registers were written in the previous time slot (where TSMn=1), the TUE status bit will be set and the hardware will operate as if the STX register had been written. The STD pin will be enabled and the contents of the STX will be transmitted again. This may lead to drive conflicts on the transmit data line, in another device is transmitting data during this time slot.
5	TXSR Register	STD Pin	On active time slots, the TXSR register contents are shifted out on the STC pin, one bit per rising edge of SCK. On inactive time slots, the STD pin is tri-stated so it can be driven by another device.

1. **Section 12.13** provides a complete description of the interrupt process.

12.5.4.2 Operation Using RSM Register

When the RSM register is included in the design, interrupt overhead can be reduced. If all bits of the RSM register are set, the ESSI receiver will continue to operate as previously described. The RSM register is used to automatically discard data from selected time slots. This is accomplished by writing the RSM with 0 in the selected time-slot bit location. This means no data is transferred from the Receive Data Shift Register (RXSR) on the zero time-slots, no status flags change, and no interrupts are generated.

Receiver timing in the Network mode, and using RSM registers for an 8-bit word with a continuous clock, results in the disabled FIFO five words per frame sync. In this example there are only two receive interrupts per frame instead of five as in the previous example where the RSM register is not used. This process is illustrated in **Figure 12-8** and explained in **Table 12-11.**

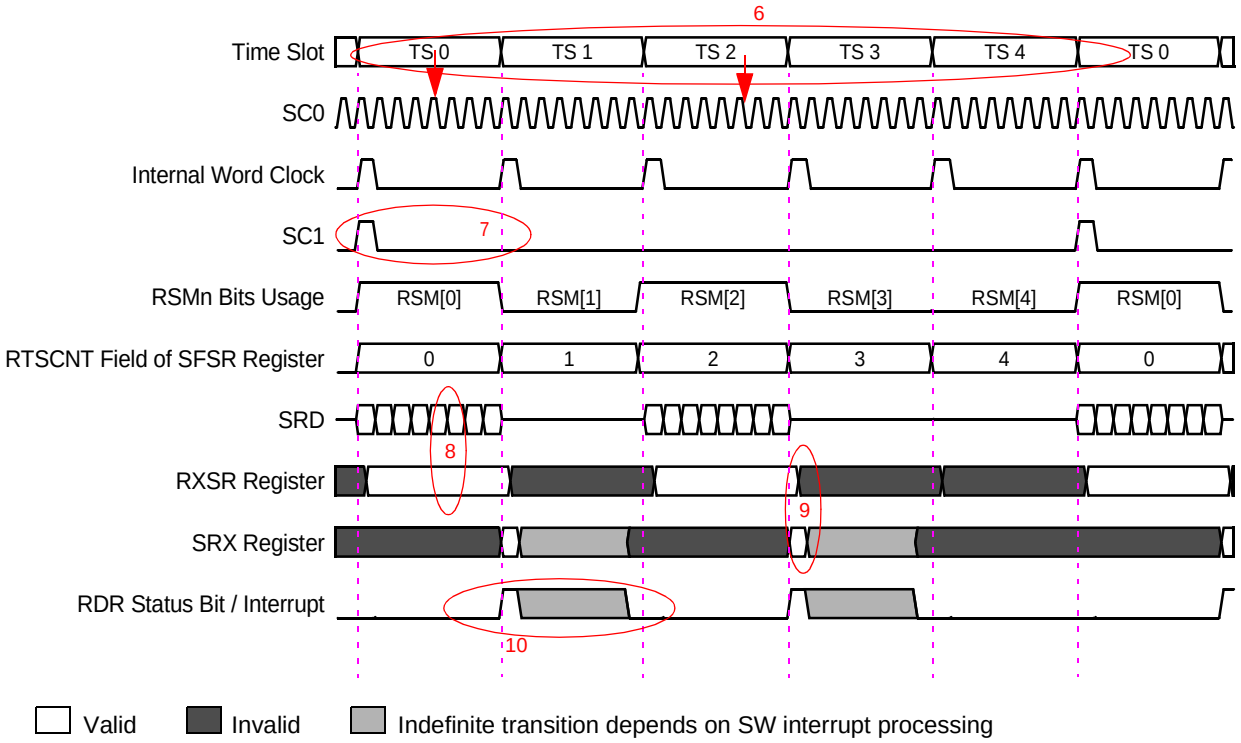


Figure 12-8. Network Mode Receive Timing with Mask Register

Table 12-11. Notes for Receive Timing with Mask Register in Figure 12-33

Note	Source Signal	Destination Signal	Description
6	—	—	Example of a five time slot frame, receive data from time slots 0 and 2. The receive hardware will only obtain data on the SRD pin when the RSMn bit is set to one.
7	SC1	—	Example with bit length frame sync and standard timing (RFSI=0, RFSL=1, and REFS=0). Frame timing begins with the rising edge of SC1.
8	SRD	RXSR Register	Data on the SRD pin is sampled on the falling edge of SC0 and shifted into the RXSR register.
9	RXSR Register	SRX Register	At the word clock, the data in the RXSR register is transferred to the SRX register, for enabled time slots.
10	RDR Status Flag and Receive Interrupt	—	This flag is set for each word clock, or time slot, where the RSMn bit is set, indicating data is available to be processed. The software must keep track of the time slots as they occur so it knows which data it is processing. When the receive interrupts are enabled (RIE=1) an interrupt will be generated when this status flag is set. The software reads the SRX register to clear the interrupt.[1]

1. **Section 12.13** provides a complete description of the interrupt process.

12.6 ESSI Configurations

Figure 12-9 and **Figure 12-10** illustrate the main ESSI configurations. These pins support all transmit and receive functions shown. **Section 12.5** describes the clock, frame sync, and data timing relationships in each of the modes available. Some modes do not require the use of all six pins. In this case, these pins can be used as MPIO pins, if desired.

Note: The MPIO is a separate module alternatively controlling the function and state of the I/O pins. See the MPIO module definition for alternate functions of the I/O pins defined here.

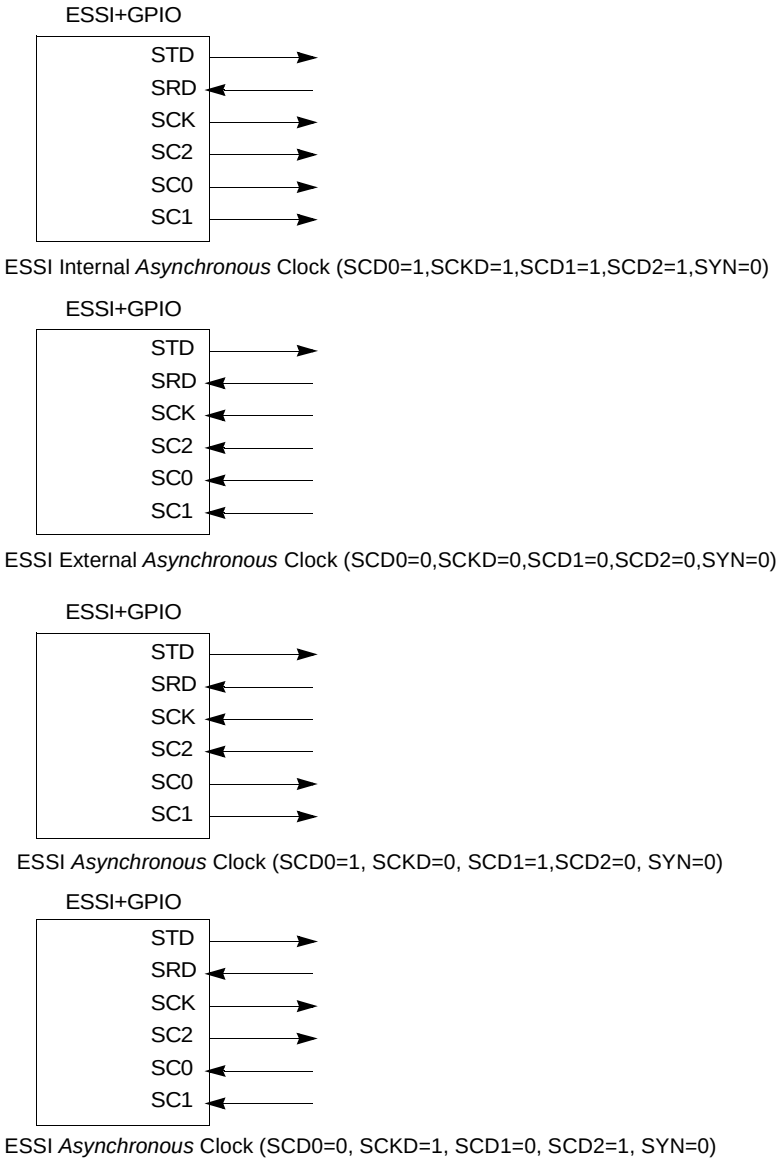
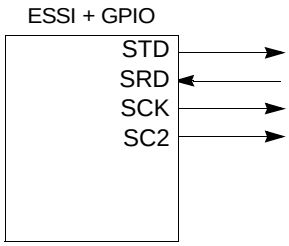
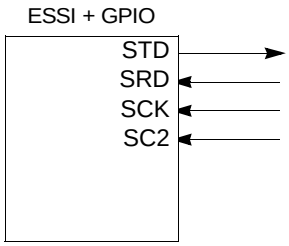


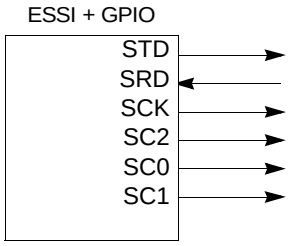
Figure 12-9. Asynchronous (SYN=0) ESSI Configurations



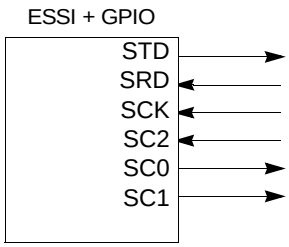
ESSI Internal *Synchronous* Clock (SCD0=X, SCKD=1, SCD1=X, SCD2=1, SYN=1)



ESSI External *Synchronous* Clock (SCD0=0, SCKD=0, SCD1=X, SCD2=0,SYN=1)



ESSI Internal *Synchronous* Clock, triple transmit (SCD0=1, SCKD=1, SCD1=1, SCD2=1, SYN=1, TE1=TE2=1)



ESSI External *Synchronous* Clock, triple transmit (SCD0=1, SCKD=0, SCD1=1, SCD2=0, SYN=1, TE1=TE2=1)

Figure 12-10. Synchronous ESSI Configurations

12.7 Module Memory Map

Nine registers make up the ESSI peripheral and are summarized in **Table 12-12** and **Figure 12-11.**

Table 12-12. ESSI Module Memory Map (ESSI0_BASE = $1FFE20, ESSI1_BASE = $1FFE00)

Address Offset	Register Acronym	Register Name	Access Type	Chapter Location
Base + $0	STX0	Transmit Data Register 0	*Write-Only*	**Section 12.8.1**
Base + $1	STX1	Transmit Data Register 1	*Write-Only*	**Section 12.8.2**
Base + $2	STX2	Transmit Data Register 2	*Write-Only*	**Section 12.8.3**
Base + $3	SRX	Receive Data Register	*Read-Only*	**Section 12.8.4**
Base + $4	SSR	Status Register	*Read-Only*	**Section 12.8.7**
Base + $5	SCR2	Control Register 2	Read/Write	**Section 12.8.8**
Base + $6	SCR3	Control Register 3	Read/Write	**Section 12.8.9**
Base + $7	SCR4	Control Register 4	Read/Write	**Section 12.8.10**
Base + $8	STXCR	Transmit Control Register	Read/Write	**Section 12.8.11**
Base + $9	SRXCR	Receive Control Register	Read/Write	
Base + $A	STSR	Time Slot Register	*Write-Only*	**Section 12.8.12**
Base + $B	SFCSR	FIFO Control/Status Register	*Read-Only*	**Section 12.8.13**
Base + $C	TSMA	Transmit Slot Mask Register	Read/Write	**Section 12.8.14**
Base + $D	TSMB	Transmit Slot Mask Register	Read/Write	
Base + $E	RSMA	Receive Slot Mask Register	Read/Write	**Section 12.8.15**
Base + $F	RSMB	Receive Slot Mask Register	Read/Write	

Add. Offset	Register Name		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
$0	STX0	R																
		W	DATA															
$1	STX1	R																
		W	DATA															
$2	STX2	R																
		W	DATA															
$3	SRX	R	HIGH BYTE								LOW BYTE							
		W																
$4	SSR	R	IF1	IF0		TFF	TLS	RLS	TF1ERR	TF2ERR	RDR	TDE	ROE	TUE	TFS	RFS	RFF	TFE
		W																
$5	SCR2	R	RIE	TIE	RE	TE0	TE1	TE2			SYN	TSHFD	TSCKP	ESSIEN	NET	TFSI	TFSL	TEFS
		W																
$6	SCR3	R	DIV4DIS	RSHFD	RSCKP	RD MAE	TD MAE	RFSI	RFSL	REFS	RFEN	TFEN	INIT			SYNRST	RLIE	TLIE
		W																
$7	SCR4	R						TXSF0	TXSF1	TXSF2		SSC1	SCKD	SCD2	SCD1	SCD0	OF1	OF0
		W																

Figure 12-11. ESSI Register Map Summary

Add. Offset	Register Name		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
$8	STXCR	R	PSR	WL		DC					PM							
		W																
$9	SRXCR	R	PSR	WL		DC					PM							
		W																
$A	STSR	R																
		W	DUMMY REGISTER WRITTEN DURING INACTIVE TIME SLOTS (NETWORK MODE)															
$B	SFCSR	R	RFCNT				TFCNT				RFWM				TFWM			
		W																
$C	TSMA	R	TSMA [15:0]															
		W																
$D	TSMB	R	TSMB [31:16]															
		W																
$E	RSMA	R	RSMA [15:0]															
		W																
$F	RSMA	R	RSMB [31:16]															
		W																

R	0	Read as 0
W		Reserved

Figure 12-11. ESSI Register Map Summary (Continued)

12.8 Register Descriptions (ESSI0_BASE = $1FFE20, ESSI1_BASE = $1FFE00)

All registers are addressed by word. Byte accesses to the registers are not supported. Eight registers are *not* accessible. Registers associated with the Enhanced SSI (ESSI) are delineated in the Memory Chapter, **Table 3-16** and **Table 3-17.**

12.8.1 ESSI Transmit Registers (STX0, STX1, STX2)

The ESSI Transmit Data (STX02) are *write-only* 16-bit registers. Data to be transmitted is written to these registers. If the Transmit FIFO Registers are utilized, data is transferred from the STX02 registers to the Transmit FIFO Register as the FIFOs have available space. Otherwise, data written to these registers is transferred to the Transmit Shift Register (STX02) whenever the transmit FIFO feature is enabled. When the feature is enabled, the Transmit Shift Registers (TXSR02) receive their values from these FIFO registers. Transmitted data is first-in-first-out. If the transmit FIFO feature is not enabled, this register is bypassed and the contents of the Transmit Data Registers (STX02) is transferred into the Transmit Shift Registers (TXSR02). When the Transmit Interrupt Enable (TIE) bit in SCR2 register is set the DSC is interrupted when the level of data in an enabled transmit FIFO falls below the selected threshold and the Transmit Data Register Empty (TDE) bit in the ESSI Status Register (SSR) is set.

When both Transmit FIFO and Transmit Data Registers are full, any further write will overwrite the content of the Transmit Data Registers (STX02).

Note: Enable ESSI (ESSIEN = 1), before writing to Transmit Data Register and Transmit FIFO.

Base + $0	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read																
Write	DATA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 12-12. ESSI Transmit Data Register 0 (STX0)

See Programmer's Sheet on Appendix page B - 68

Base + $1	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read																
Write	DATA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 12-13. ESSI Transmit Data Register 1 (STX1)

See Programmer's Sheet on Appendix page B - 68

Base + $2	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read																
Write	DATA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 12-14. ESSI Transmit Data Register 2 (STX2)

See Programmer's Sheet on Appendix page B - 68

12.8.2 ESSI Transmit FIFO Registers (TXFIFO0, TXFIFO1, TXFIFO2)

The 8 × 16-bit TXFIFO registers are used to buffer samples written to the Transmit Data Registers (STX0-2). They are written by the contents of the Transmit Data Registers (STX0-2) whenever the transmit FIFO feature is enabled. If enabled, the Transmit Shift Registers (TXSR0-2) receive their values from the FIFO registers. If the transmit FIFO feature is not enabled, this register is bypassed and the contents of STX0-2 registers are transferred into the Transmit Shift Registers (TXSR0-2).

When the Transmit Interrupt Enable (TIE) bit in the SCR2 and Transmit Data Register Empty (TDE) bit in the SSR are set, the transmit interrupt is asserted whenever STX0-2 are empty and the data level in the ESSI transmit FIFO falls below the selected threshold.

When both TXFIFO and STX are full, any further write will overwrite the content of TXFIFO and STX.

Note: Enable ESSI before writing to TXFIFO and STX.

12.8.3 ESSI Transmit Shift Registers (TXSR0, TXSR1, TXSR2)

The TXSR0-2 are 16-bit shift registers containing transmitted data. Data is shifted out to the Serial Transmit Data (STD, SC0, and SC1) pins by the selected internal/external bit clock when the associated internal/external frame sync is asserted. The Word Length (WL) control bits in the ESSI Transmit Control Register (STXCR) determine the number of bits to be shifted out of the TXSR0-2 registers before it is considered empty and can receive further writing. Please refer to **Section 12.8.11** for more information. This word length can be 8, 10, 12, or 16 bits. The data to be transmitted occupies the most significant portion of the Shift register. The unused portion of the register is ignored. Data is always shifted out of this register with the Most Significant Bit (MSB) first when the TSHFD bit of the SCR2 is cleared, illustrated in **Figure 12-15.** If this bit is set, the Least Significant Bit (LSB) is shifted out first, diagramed in **Figure 12-16.**

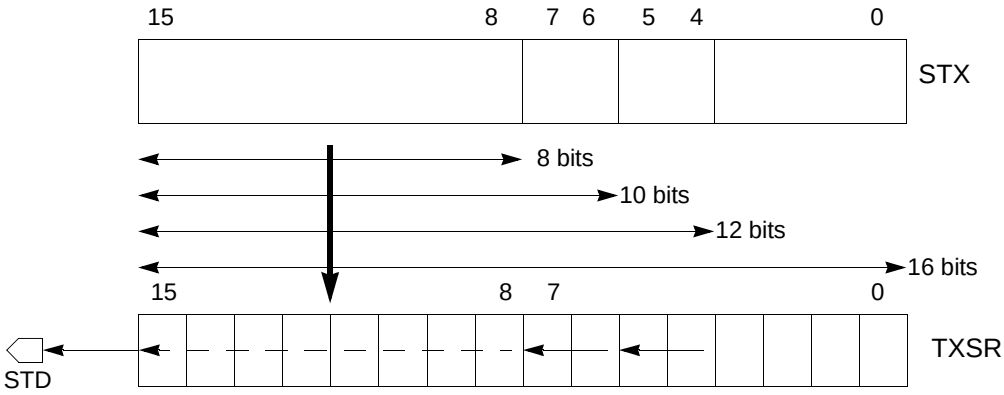


Figure 12-15. Transmit Data Path (TSHFD = 0)

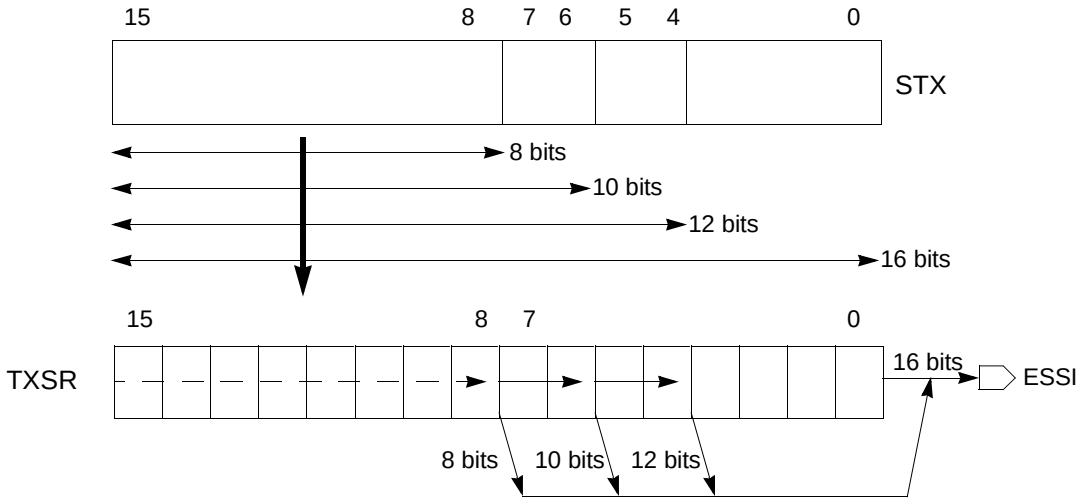


Figure 12-16. Transmit Data Path (TSHFD = 1)

12.8.4 ESSI Receive Register (SRX)

The SRX is a 16-bit, *read-only* register. It always accepts data from the Receive Shift Register (RXSR) as it becomes full. The data read occupies the most significant portion of the SRX register. The unused bits (least significant portion) are read as 0s.

If the Receive Data Full Interrupt is enabled, the interrupt is asserted whenever the SRX register becomes full. When the Receive FIFO is also enabled, the Receive FIFO must be above its watermark before the interrupt is asserted.

Base + $3	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	HIGH BYTE								LOW BYTE							
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 12-17. ESSI Receive Data Register (SRX)

See Programmer's Sheet on Appendix page B - 69

12.8.5 ESSI Receive FIFO Register (RXFIFO)

The RXFIFO is a 8×16-bit FIFO register used to buffer samples received in the ESSI Receive Data Shift Register (RXSR). The receive FIFO is enabled by setting the RFEN bit of the SCR3 Control Register.

Received data is then held in the FIFO if the data in the SRX has not yet been read.

If the receive interrupt is enabled, it is asserted whenever the SRX is full and the data level in the ESSI receive FIFO reaches the selected threshold. If the receive FIFO feature is not enabled, this register is bypassed and the Receive Shift Register (RXSR) data is automatically transferred into the SRX.

12.8.6 ESSI Receive Shift Register (RXSR)

This is a 16-bit shift register receiving incoming data from the serial receive data SRD pin. Data is shifted in by the selected internal/external bit clock when the associated internal/external frame sync is asserted. Data is assumed to be received MSB first if the RSHFD bit of the SCR3 is cleared, illustrated in **Figure 12-18.** If this bit is set, the data is received LSB first, as diagramed in **Figure 12-19.** Data is transferred to the ESSI Receive Data Register (SRX) or receive FIFO, if the receive FIFO is enabled and SRX is full, after 8, 10, 12, or 16 bits have been shifted in depending on the WL control bits. When receiving 8, 10, or 12 bits data, LSB bits are set to 0.

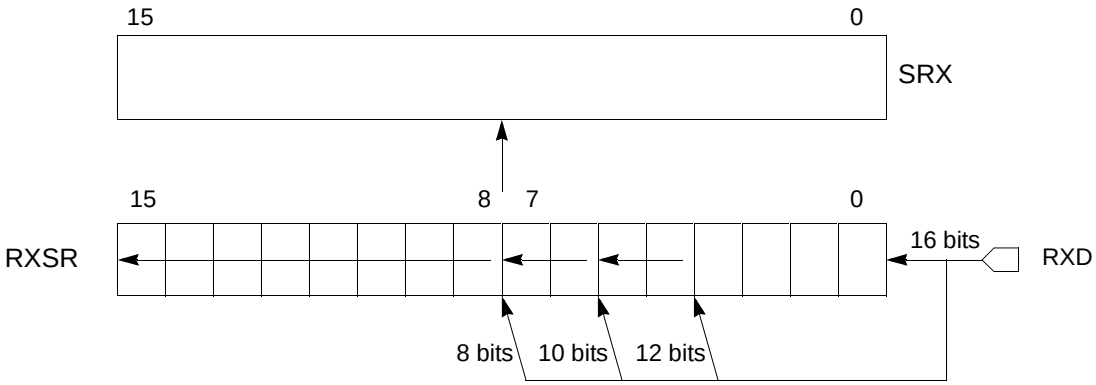


Figure 12-18. Receive Data Path (RSHFD = 0)

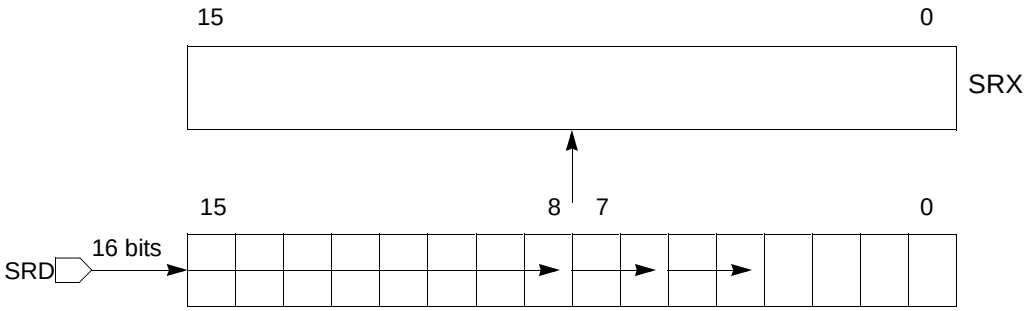


Figure 12-19. Receive Data Path (RSHFD = 1)

12.8.7 ESSI Status Register (SSR)

This is a 16-bit, *read-only* register except for the RLS and TLS bits described in the following third note. It is used to monitor the ESSI. The register is used to interrogate the status and serial input flags of the ESSI. The status bits are described in the following paragraphs.

Base + $4	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	IF1	IF0		TFF	TLS	RLS	TF1ERR	TF2ERR	RDR	TDE	ROE	TUE	TFS	RFS	RFF	TFE
Write																
Reset	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1

Figure 12-20. ESSI Status Register (SSR)

See Programmer's Sheet on Appendix page B - 70

Note: ESSI Status flag is updated when the ESSI is enabled.

Note: All flags in the status portion of the SSR are updated after the first bit of the next ESSI word has completed transmission or reception. The ROE and TUE status bits are cleared by reading the SSR followed by a read/write to either the SRX or STX0-2 register, respectively.

Note: The RLS and TLS status bits are cleared by reading the SSR, then writing 1 to the appropriate bit of the SSR.

12.8.7.1 Input Flag 1 (IF1)—Bit 15

The IF1 bit is enabled only when SC1 is configured as an input flag and the Synchronous mode is selected. For example, when the SYN bit is set and the TE2 and SCD1 bits are cleared. Please refer to **Table 12-15.**

The ESSI latches any data present on the SC1 signal during reception of the first received bit after the frame sync is detected. The IF1 bit is updated with this data when the data in the Receive Shift Register is transferred into the Receive Data Register

If it is not enabled, the IF1 bit is cleared.

12.8.7.2 Input Flag 0 (IF0)—Bit 14

The IF0 bit is enabled only when SC0 is configured as an input flag and the Synchronous mode is selected; that is to say, when the SYN bit is set and the TE1 and SCD0 bits are cleared. Please see **Table 12-16** for more information.

The ESSI latches any data present on the SC0 signal during reception of the first received bit after the frame sync is detected. The IF0 bit is updated with this data when the data in the Receive

Shift Register is transferred into the Receive Data Register. If it is not enabled, the IF0 bit is cleared.

12.8.7.3 Reserved—Bit 13

This bit is reserved or not implemented. It cannot be read nor modified by writing.

12.8.7.4 Transmit FIFO Full (TFF)—Bit 12

This status bit allows monitoring when the Transmit FIFO is full. The state of this bit reflects the status of the transmitter(s) selected by the TXSF0-2 control bits.

- 0 = Transmit FIFO can accept more data
- 1 = Transmit FIFO is full

12.8.7.5 Transmit Last Slot (TLS)—Bit 11

This is a status bit indicating the timing of the last transmit slot during the Network mode operation. When this bit is set, the Transmit Last Slot Interrupt is asserted. The interrupt service routine for this interrupt should read the Status Register, then write 1 to this bit to clear the interrupt.

If the Transmit Last Slot Interrupt is not enabled, this bit can be read by the software to determine the timing of the last slot. When the TLIE bit in the SCR3 register is disabled the status bit will only be asserted during the last slot timing.

- 0 = Not currently transmitting the last time slot of the transmit frame
- 1 = Last slot of the transmit frame is currently being transmitted

12.8.7.6 Receive Last Slot (RLS)—Bit 10

This is a status bit indicating the timing of the last receive slot during the Network mode operation. When this bit is set, the Receive Last Slot (RLS) interrupt bit is asserted. The interrupt service routine for this interrupt should read the Status Register and then write 1 to this bit to clear the interrupt.

If the RLS interrupt is not enabled, this bit can be read by the software to determine the timing of the last slot. When the RLIE bit in the SCR3 register is disabled the status bit will only be asserted during the last slot timing.

- 0 = Not currently receiving the last time slot of the receive frame
- 1 = Last slot of the receive frame is currently being received

12.8.7.7 Transmit FIFO 1 Error (TF1ERR)—Bit 9

When the transmitter status control TXSF1 is set and FIFOs are in use, this status bit will indicate the state of FIFO 1 is not the same as FIFO 0. If Transmitter 0 (TXSF0 = 0) is not in use this flag can never be set.

- 0 = State of TXFIFO0 and TXFIFO1 are the same (contain the same amount of data)
- 1 = State of TXFIFO0 is different than the state of TXFIFO1

12.8.7.8 Transmit FIFO 2 Error (TF2ERR)—Bit 8

When the transmitter status control TXSF2 is set and FIFOs are in use, this status bit indicates the state of FIFO2 is not the same as FIFO0. If Transmitter 0 (TXSF0 = 0) is not in use, this status bit will indicate the state of FIFO2 is not the same as FIFO1. If Transmitter 1 is also not operating (TXSF0 = TXSF1 = 0), this flag can never be set.

- 0 = Status of TXFIFO2 matches the other enabled TXFIFOs
- 1 = Status (data content level) of TXFIFO2 is different than the other enabled TXFIFOs

12.8.7.9 Receive Data Ready Flag (RDR)—Bit 7

This flag bit is set when Receive Data Register (SRX) or receive FIFO (RXFIFO) is loaded with a new value. RDR is cleared when the CPU reads the SRX register. If RXFIFO is enabled, RDR is cleared when receive FIFO is empty.

If the RIE bit is set, a receive data interrupt request is issued when the RDR bit is set. The interrupt request vector depends on the state of the Receiver Overrun Error (ROE) bit on the SSR. The RDR bit is cleared by Power-On Reset (POR) and ESSI reset (ESSIEN = 0).

12.8.7.10 Transmit Data Register Empty (TDE)—Bit 6

This flag bit is set when there is no data waiting to be transferred to the TXSR register. A transmit FIFO (TXFIFO) is enabled when there is at least one empty slot in STX or TXFIFO. When the TXFIFO is not enabled, the STX is empty. For example, when the contents of the STX register are transferred into the Transmit Shift Register (TXSR). When set, the TDE bit indicates data should be written to the STX register or to the STSR before the Transmit Shift Register becomes empty, or an underrun error will occur.

The TDE bit is cleared when data is written to the STX register or to the STSR to disable transmission of the next time slot. If the TIE bit is set, an ESSI transmit data interrupt request is issued when the TDE bit is set. The vector of the interrupt depends on the state of the TUE bit in the SSR. The TDE bit is set by Power-On Reset (POR) and ESSI reset (ESSIEN = 0).

12.8.7.11 Receive Overrun Error (ROE)—Bit 5

This flag bit is set when the Receive Shift Register (RXSR) is enabled, filled, ready to transfer to the SRX, or the RXFIFO registers, and when these registers are already full. If the Receive FIFO is enabled, it is indicated by the Receive FIFO Full (RFF) bit otherwise this is indicated by the Receive Data Ready (RDR) bit being set. The RXSR is not transferred in this case.

Note: When using the RXFIFO with a watermark other than eight, the ROE bit does not mean data has been lost. The RXCNT field of the SFCSR should be checked to determine the likelihood of actual data loss.

A Receive Overrun Error (ROE) does not cause interrupts. However, when the ROE bit is set, it causes a change in the interrupt vector used, allowing the use of a different interrupt handler for a receive overrun condition. If a receive interrupt occurs with the ROE bit set, the receive data with exception status interrupt is generated. If a receive interrupt occurs with the ROE bit cleared, the Receive Data Interrupt is generated. The ROE bit is cleared by Power-On Reset (POR) or ESSI reset (ESSIEN = 0) and by reading the SSR with the ROE bit set followed by reading the SRX register. Clearing the RE bit does not affect the ROE bit.

12.8.7.12 Transmitter Underrun Error (TUE)—Bit 4

This flag bit is set when the TXSR is empty, or when there is no data to be transmitted, indicated by the TDE bit being set and a transmit time slot occurs. When a Transmit Underrun Error occurs, the previously sent data is retransmitted.

A transmit time-slot in the Normal mode occurs when the frame sync is asserted. Each time-slot requires data transmission in the Network mode, it may cause a TUE error.

The TUE bit does not cause interrupts. However, the TUE bit will cause a change in the interrupt vector used for transmit interrupts. Consequently, a different interrupt handler can be used for a transmit underrun condition. If a transmit interrupt occurs with the TUE bit set, the transmit data with exception status interrupt is generated. If a transmit interrupt occurs with the TUE bit cleared, the transmit data interrupt is generated.

The TUE bit is cleared by Power-On Reset (POR) or ESSI reset (ESSIEN = 0). The TUE bit is also cleared by reading the SSR with the TUE bit set, followed by writing to the STX register or to the STSR.

The state of this bit reflects the status of the transmitter(s) selected by the TXSF0-2 control bits in the SCR4 register.

12.8.7.13 Transmit Frame Sync (TFS)—Bit 3

The middle section of **Figure 12-21**exhibits data written to the STX Register in the Network mode during the time slot when the TFS bit is set. This is transmitted during either of the following:

- Second time slot (in the Network mode)
- In the following first time slot (in the Normal mode)

While in Network mode, the TFS bit is set during transmission of the first slot of the frame. The bit is then cleared when starting transmission of the next slot. The TFS bit is cleared by Power-On Reset (POR) or ESSI reset (ESSIEN = 0).

12.8.7.14 Receive Frame Sync (RFS)—Bit 2

When set, this flag bit indicates a frame sync occurred during receiving of the next word into the SRX Register. The process is illustrated in the lower portion of **Figure 12-21.** In the Network mode, the RFS bit is set while the first slot of the frame is being received. It is cleared when the next slot of the frame begins to be received. The RFS bit is cleared by Power-On Reset (POR) or ESSI reset (ESSIEN = 0).

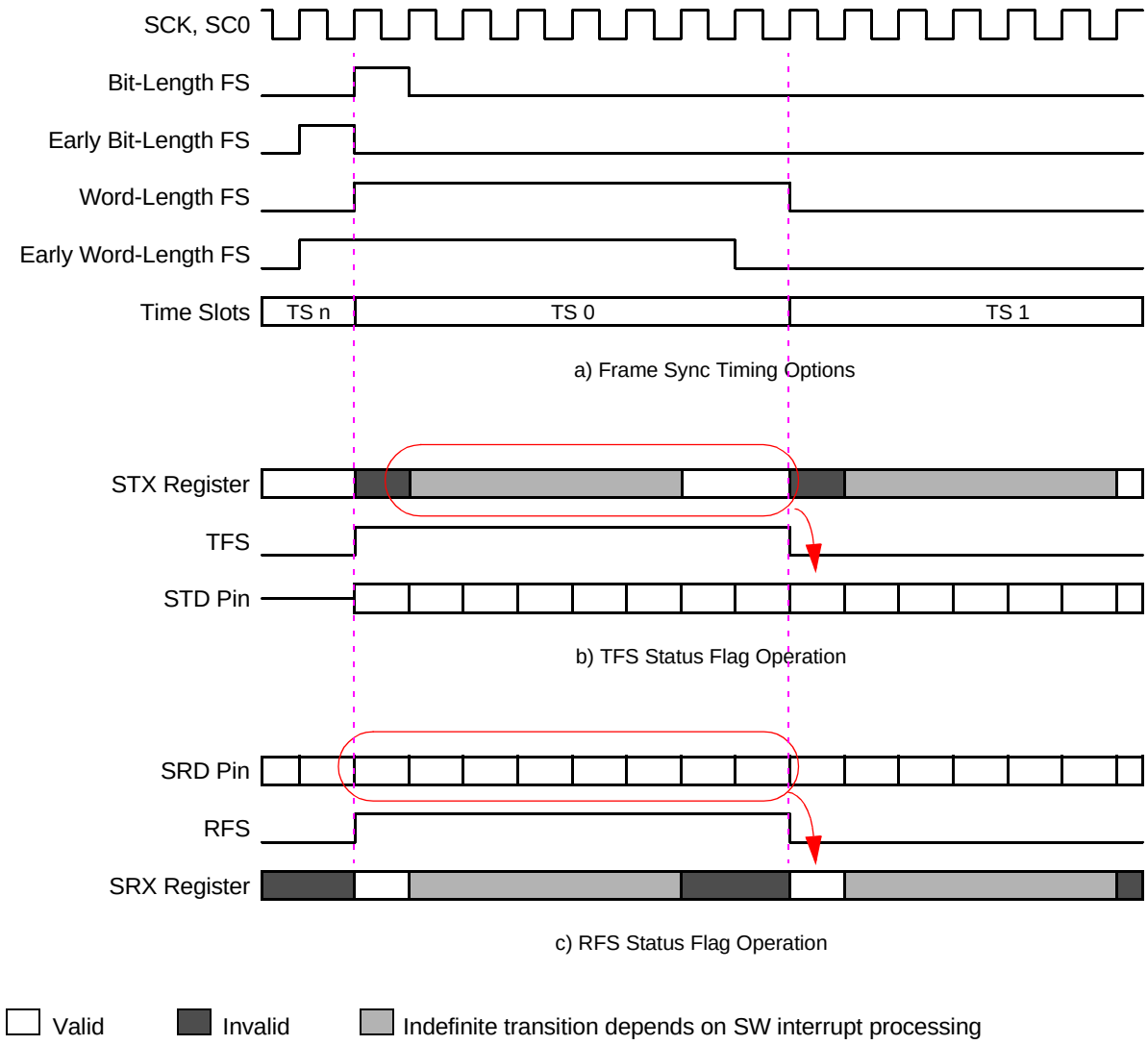


Figure 12-21. Frame Sync Timing Options

12.8.7.15 Receive FIFO Full (RFF)—Bit 1

This flag bit is set when the *receive* section is programmed with the enabled Receive FIFO. The data level in the RXFIFO must reach the selected Receive FIFO Watermark (RFWM) threshold. When set, the RFF indicates data can be read using the SRX register.

Note: An interrupt is generated only if both RFF and RIE bits are set and if the RXFIFO is enabled.

The RFF bit is cleared in normal operation by reading the SRX register. The RFF is cleared by Power-On Reset (POR) or disabling the ESSI (ESSIEN = 0). When the RXFIFO is completely full, all further received data is ignored until current data is read.

12.8.7.16 Transmit FIFO Empty (TFE)—Bit 0

This flag bit is set when the *transmit* section is programmed with an enabled TXFIFO and the data level in the TXFIFO falls below the selected Transmit FIFO Watermark (TFWM) threshold. When set, the TFE bit indicates data can be written to the TXFIFO register. The TFE bit is cleared by writing data to the STX register until the TXFIFO data content level reaches the watermark level.

Note: An interrupt is generated only if both TFE and the TIE bits are set if a transmit FIFO is enabled.

The TFE bit is set by Power-On Reset (POR) and when ESSI is disabled (ESSIEN = 0).

12.8.8 ESSI Control Register 2 (SCR2)

The ESSI Control Register 2 (SCR2) is one of five 16-bit, read/write control registers used to direct the operation of the ESSI. The ESSI reset is controlled by a bit in SCR2. Interrupt enable bits for the receive and transmit sections are provided in this control register. ESSI operating modes are also selected in this register.

The Power-On Reset (POR) clears all SCR2 bits. The ESSI reset (ESSIEN = 0) does not affect the SCR2 bits. The SCR2 bits are described in the following paragraphs.

Note: As with all on-chip peripheral interrupts, the interrupt controller must be configured to allow ESSI maskable interrupts and determine their interrupt priority level, before enabling them in this register.

Base + $5	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	RIE	TIE	RE	TE0	TE1	TE2			SYN	TSHFD	TSCKP	ESSIEN	NET	TFSI	TFSL	TEFS
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 12-22. ESSI Control Register 2 (SCR2)

See Programmer’s Sheet on Appendix page B - 72

12.8.8.1 Receive Interrupt Enable (RIE)—Bit 15

This control bit allows interrupting the program controller. When the RIE and RE bits are set, the program controller is interrupted when the ESSI receives data. As provided in **Table 12-13,** the interrupt trigger depends on whether the receive FIFO is enabled.

If the receive FIFO is disabled:

- 0 = No interrupt is generated.
- 1 = An interrupt is generated when the RDR flag in the SSR is set. One value can be read from the SRX register. Reading the SRX register clears the RDR bit, thus clearing the interrupt.

If the receive FIFO is enabled:

- 0 = No interrupt is generated.
- 1 = An interrupt is generated when the RFF flag in the SSR is set. A maximum of eight values are available to be read from the SRX register. To clear the RFF bit interrupt, read the SRX register, removing data from the Receive FIFO, thereby dropping the data level below the watermark.

Two receive data interrupts with separate interrupt vectors are available:

1. Receive data with exception status.
2. Receive data without exception.

Generated interrupt vectors and their conditions are listed in **Table 12-13.**

Table 12-13. ESSI Receive Data Interrupts [1]

Interrupt	RIE	Selection Control		ROE
		RFEN = 0	RFEN = 1	
Receive Data with Exception Status	1	RDR = 1	RFF = 1	1
Transmit Date without Exception Status	1	RDR = 1	RFF = 1	0

1. **Table 12-27** provides a complete list of interrupts.

12.8.8.2 Transmit Interrupt Enable (TIE)—Bit 14

This control bit provides program controller interruption. When the TIE and TE0-2 bits are set the program controller is interrupted when the ESSI needs more transmit data lists. The interrupt trigger depends whether the transmit FIFOs are enabled, shown in **Table 12-14.**

If the transmit FIFO is disabled:

- 0 = No interrupt is generated.
- 1 = An interrupt is generated when the TDE flag in the SSR is set. One value can be written to each of the enabled STX0-2 registers when this interrupt occurs.

If the transmit FIFO is enabled:

- 0 = No interrupt is generated.
- 1 = An interrupt is generated when the TFE flag in the SSR is set. When this interrupt occurs, up to eight values can be written to each of the enabled STX0-2 registers, depending on the level of the TXFIFO watermark.

The TDE bit always indicates the STX register empty condition even when the transmitter is disabled by the Transmit Enable (TE) bits in the SCR2. Writing data to the STX register or STSR clears the TDE bit, thus clearing the interrupt. Two transmit data interrupts with separate interrupt vectors are available:

1. Transmit data with exception status
2. Transmit data without exceptions

12.8.8.3 **Table 12-14** lists the conditions these interrupts are generated.

Table 12-14. ESSI Transmit Data Interrupts [1]

Interrupt	TIE	Selection Control		TUE
		TFEN = 0	TFEN = 1	
Transmit Data with Exception Status	1	TDE = 1	TFE = 1	1
Transmit Date without Exception Status	1	TDE = 1	TFE = 1	0

1. **Table 12-27** provides a complete list of interrupts.

12.8.8.4 Receive Enable (RE)—Bit 13

This control bit enables the receive portion of the ESSI.

- 0 = Receiver is disabled by inhibiting data transfer into the SRX. If data is being received when this bit is cleared, the rest of the word is not shifted in, or is it transferred to the SRX register.
- 1 = Receive portion of the ESSI is enabled and received data will be processed starting with the next receive frame sync.

If the RE bit is re-enabled during a time slot before the second to last bit, then the word will be received. It is recommended to clear this bit when clearing ESSIEN.

12.8.8.5 Transmit Enable 0 (TE0)—Bit 12

This control bit enables the transfer of the contents of the STX0 register to its Transmit Data Shift Register 0 (TXSR0). TE0 is functional when the ESSI is in either Synchronous or Asynchronous modes.

- 0 = The transmitter continues to send the data currently in TXSR0, then disables the transmitter.
- 1 = On the next frame boundary, the transmit 0 portion of the ESSI is enabled. With internally generated clocks, the frame boundary will occur within a word time. If the TE0 bit is cleared, then set again during the same transmitted word, the data continues to be transmitted. If the TE0 bit is set again during a different time slot, data is not transmitted until the next frame boundary.

The serial output is tri-stated and any data present in the STX0 register is not transmitted. For example, data can be written to the STX0 register with the TE0 bit cleared and the TDE bit is cleared, but data is not transferred to the TXSR0.

The Normal Transmit Enable sequence is to write data to the STX0 register or to the STSR before setting the TE0 bit. The Normal Transmit Disable sequence is to clear both the TE0 and TIE bits after the TDE bit is set. This bit should be cleared when clearing ESSIEN.

12.8.8.6 Transmit Enable 1 (TE1)—Bit 11

This control bit enables the transfer of the contents of the STX1 register to its Transmit Data Shift Register 1 (TXSR1). TE1 is functional when the ESSI is in the Synchronous mode. It is ignored when the ESSI is in the Asynchronous mode.

- 0 = The transmitter continues to send the data currently in TXSR1, then disables the transmitter. The serial output is tri-stated and any data present in the STX1 register is not transmitted. In other words, data can be written to the STX1 register with the TE1 bit cleared and the TDE bit is cleared but data is not transferred to the TXSR1.
- 1 = On the next frame boundary, the Transmit 1 portion of the ESSI is enabled for that frame. With internally generated clocks, the frame boundary will occur within a word time. If the TE1 bit is cleared and then set again during the same transmitted word, the data continues to be transmitted. If the TE1 bit is set again during a different time slot, data is not transmitted until the next frame boundary.

When the TE1 bit remains clear until the beginning of the next frame, it causes the SC0 signal to act as a serial I/O flag from the start of the frame in both Normal and Network modes.

The Normal Transmit Enable sequence is to write data to the STX1 register or to the STSR before setting the TE1 bit. The Normal Transmit Disable sequence is to clear the TE1 and TIE bits after the TDE bit is set. This bit should be cleared when clearing ESSIEN.

Note: Setting the TE1 bit does not affect the generation of frame sync or output flags.

12.8.8.7 Transmit Enable 2 (TE2)—Bit 10

This control bit enables the transfer of the contents of the STX2 register to its Transmit Data Shift Register 2 (TXSR2). The TE2 bit is functional when the ESSI is in the Synchronous mode. It is ignored when the ESSI is in the Asynchronous mode.

- 0 = The transmitter continues to send the data currently in TXSR2, then disables the transmitter. The serial output is tri-stated and any data present in the STX2 register is not transmitted; that is to say, data can be written to the STX2 register with the TE2 and TDE bits cleared, but data is not transferred to the TXSR2.
- 1 = On the next frame boundary, the transmit 2 portion of the ESSI is enabled for that frame. With internally generated clocks, the frame boundary will occur within a word time. If the TE2 bit is cleared, then set again during the same transmitted word, the data continues to be transmitted. If the TE2 bit is set again during a different time slot, data is not transmitted until the next frame boundary.

When the TE2 bit remains clear until the start of the next frame, it causes the SC1 signal to act as a serial I/O flag from the start of the frame in both Normal and Network modes.

The Normal Transmit Enable sequence is to write data to the STX2 register or to the STSR before setting the TE2 bit. The Normal Transmit Disable sequence is to clear the TE2 bit and the TIE bit after the TDE bit is set. This bit should be cleared when clearing ESSIEN.

Note: Setting the TE2 bit does not affect the generation of frame sync or output flags.

12.8.8.8 Reserved—Bits 9–8

These bits are reserved or not implemented. They cannot be read nor modified by writing.

12.8.8.9 Synchronous Mode (SYN)—Bit 7

This control bit enables the Synchronous mode of operation. In this mode, the transmit and receive sections share a common clock pin (SCK) and frame sync pin (SC2). **Table 12-3** illustrates the clock configuration options. **Table 12-4** illustrates the frame sync options.

Note: Synchronous mode operation with all transmitters disabled and the receiver enabled requires an externally generated frame sync.

12.8.8.10 Transmit Shift Direction (TSHFD)—Bit 6

This bit controls whether the MSB or LSB is transmitted first for the transmit section.

- 0 = MSB is transmitted first.
- 1 = LSB is transmitted first.

Note: The codec device labels the MSB as bit 0, whereas the ESSI labels the LSB as bit 0. Therefore, when using a standard codec, the ESSI MSB (or codec bit 0) is shifted out first, and the TSHFD bit should be cleared.

12.8.8.11 Transmit Clock Polarity (TSCKP)—Bit 5

This control bit determines which bit clock edge is used to clock out data in the transmit section.

- 0 = Data is clocked out on the rising edge of the bit clock.
- 1 = The falling edge of the bit clock is used to clock the data out.

12.8.8.12 ESSI Enable (ESSIEN)—Bit 4

This control bit enables and disables the ESSI.

- 0 = The ESSI is disabled and held in a reset condition. When disabled, all output pins are tri-stated, the status register bits are preset to the same state produced by the Power-On Reset (POR) and the Control register bits are unaffected. The contents of the STX, TXFIFO and RXFIFO are cleared when this bit is reset. When ESSI is disabled, all internal clocks are disabled except clocks required for register access. When clearing ESSIEN, it is recommended to also clear RE and TE0-2.
- 1 = The ESSI is enabled, causing an output frame sync to be generated when set up for internal frame sync, or it causes the ESSI to wait for the input frame sync when set up for external frame sync.

12.8.8.13 Network Mode (NET)—Bit 3

This control bit selects the operational mode of the ESSI.

- 0 = Normal mode is selected.
- 1 = Network mode is selected.

12.8.8.14 Transmit Frame Sync Invert (TFSI)—Bit 2

This control bit selects the logic of frame sync I/O.

- 0 = Frame sync is active high
- 1 = Frame sync is active low

12.8.8.15 Transmit Frame Sync Length (TFSL)—Bit 1

This control bit selects the length of the frame sync signal to be generated or recognized. See the upper portion of **Figure 12-21** for an example timing diagram of the frame sync options.

- 0 = A one-word long frame sync is selected. The length of this word-long frame sync is the same as the length of the data word selected by WL.
- 1 = A one-bit long frame sync is selected

The frame sync is deasserted after one bit-for-bit length frame sync and after one word- for-word length frame sync.

12.8.8.16 Transmit Early Frame Sync (TEFS)—Bit 0

This bit controls when the frame sync is initiated for the transmit and receive sections. Please refer to the upper portion of **Figure 12-21** for a timing diagram example of the frame sync options.

- 0 = Frame sync is initiated as the first bit of data is transmitted.
- 1 = Frame sync is initiated one-bit before the data is transmitted. The frame sync is disabled after one bit-for-bit length frame sync and after one word-for-word length frame sync.

12.8.9 ESSI Control Register 3 (SCR3)

The ESSI Control Register 3 (SCR3) is one of five 16-bit, read/write control registers used to direct the operation of the ESSI. SCR3 controls enabling of the last slot interrupts for Network mode transmit and receive operations.

The Power-On Reset (POR) clears all SCR3 bits. The ESSI reset (ESSIEN = 0) does not affect the SCR3 bits.

Base + $6	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	DIV4DIS	RSHFD	RSCKP	RDMAE	TDMAE	RFSI	RFSL	REFS	RFEN	TFEN	INIT			SYNRST	RLIE	TLIE
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 12-23. ESSI Control Register 3 (SCR3)

See Programmer's Sheet on Appendix page B - 75

12.8.9.1 Divider 4 Disable (DIV4DIS)—Bit 15

- 0 = FIX_CLK = IP_CLK/4 for both transmitter and receiver clock generator circuits
- 1 = FIX_CLK is equal to IP_CLK

12.8.9.2 Receive Shift Direction (RSHFD)—Bit 14

This bit controls whether the MSB or LSB is received first for the receive section.

- 0 = Data is received MSB first
- 1 = Data is received LSB first

Note: The codec device labels the MSB as bit 0, whereas the ESSI labels the LSB as bit 0. Therefore, when using a standard codec, the ESSI MSB (or codec bit 0) is shifted in first. The RSHFD bit should be cleared.

12.8.9.3 Receive Clock Polarity (RSCKP)—Bit 13

This bit controls which bit clock edge is used to capture data for the receive section.

- 0 = Captured data during clock falling edge
- 1 = Capture data input during rising edge of the clock

12.8.9.4 Receive DMA Enable (RDMAE)—Bit 12

This bit allows the ESSI to request a Direct Memory Access (DMA) module transfer of received data.

If the receive FIFO is *disabled*:

- 0 = No DMA transfer is requested.
- 1 = A DMA request is generated when the ESSI Receive Data Ready (RDR) bit is set.

If the receive FIFO is *enabled*:

- 0 = No DMA transfer is requested.
- 1 = A DMA request is generated when the ESSI Receive FIFO Full (RFF) bit in the SSR is set. The Receive FIFO Full Watermark should be set to RFWM = 01 because any other setting of RFWM can leave data in the FIFO not transferred by the DMA.

Receive Interrupt Enable (RIE) has higher priority than Receive DMA Enable (RDMAE). That is to say, if RIE is set, an interrupt is generated to the CPU instead of DMA.

Note: If DAM is not supported on the chip, this bit has no meaning and should always be cleared.

12.8.9.5 Transmit DMA Enable (TDMAE)—Bit 11

This bit allows the ESSI to request a DMA transfer for the next data to transmit.

If the transmit FIFO is *disabled*:

- 0 = No DMA transfer is requested.
- 1 = A DMA request is generated when the ESSI Transmit Data Empty (TDE) bit is set.

If the transmit FIFO is *enabled*:

- 0 = No DMA transfer is requested.
- 1 = A DMA request is generated when the ESSI Transmit FIFO Empty (TFE) bit in the SSR is set. The Transmit FIFO Empty Watermark bit can be set to any level. The FIFO is then filled to the indicated level by the DMA controller so data is readily available in the ESSI.

Transmit Interrupt Enable (TIE) has higher priority than Transmit DMA Enable (TDMAE). For example, if TIE is set, an interrupt is generated to the CPU instead of DMA.

Note: If more than one transmit channel is enabled, a DMA channel is required for each transmitter.

Note: If DMA is not supported on the chip, this bit has no meaning and should always be cleared.

12.8.9.6 Receive Frame Sync Invert (RFSI)—Bit 10

This bit selects the logic of frame sync I/O for the receive section.

- 0 = Frame sync is active high.
- 1 = Frame sync is active low.

12.8.9.7 Receive Frame Sync Length (RFSL)—Bit 9

This bit selects the length of the frame sync signal to be generated or recognized for the receive section.

Please note the first portion of **Figure 12-21** for an example timing diagram of the frame sync options.

- 0 = A one-word long frame sync is selected. The length of a word-long frame sync is the same as the length of the data word selected by WL.
- 1 = A one-bit long frame sync is selected.

12.8.9.8 Receive Early Frame Sync (REFS)—Bit 8

This bit controls when the frame sync is initiated for the receive section. Please note the first portion of **Figure 12-21** for an example timing diagram of the frame sync options.

- 0 = When the REFS bit is cleared, the frame sync is initiated as the first bit of data is received.
- 1 = The frame sync is initiated one-bit before the data is received. The frame sync is disabled after one bit-for-bit length frame sync and after one word-for-word length frame sync.

12.8.9.9 Receive FIFO Enable (RFEN)—Bit 7

This control bit enables the FIFO register for the receive section.

- 0 = The FIFO register is not used, and an interrupt request is generated (assuming interrupts are enabled) when a single sample is received by the ESSI and RDR is set.
- 1 = Allows eight samples, depending on the Receive Watermark set in the SFCSR, to be received by the ESSI. A ninth sample can be shifting in before the RFF bit is set and an interrupt request generated when enabled by the RIE bit.

12.8.9.10 Transmit FIFO Enable (TFEN)—Bit 6

This control bit enables the FIFO registers for the transmit section.

- 0 = The FIFO register is not used.
- 1 = A maximum of eight samples can be written to the STX register. A ninth sample can be shifting out.

12.8.9.11 Initialize State Machine (INIT)—Bit 5

This bit is used to initialize the state machine to reset state.

- 0 = The state machine is allowed to operate.
- 1 = Reset the TX and RX state machines. Setting this bit must be followed by a write of 0 before the state machine will operate.

12.8.9.12 Reserved—Bit 4–3

These bits are reserved or not implemented. They cannot be read or modified by writing.

12.8.9.13 Frame Sync Reset (SYNRST)—Bit 2

- 0 = Data must be read to be cleared from the registers.
- 1 = Resets the accumulation of data in the SRX register and RXFIFO bit in frame synchronization.

12.8.9.14 Receive Last Slot Interrupt Enable (RLIE)—Bit 1

- 0 = The Receive Last Slot Interrupt is disabled.
- 1 = An interrupt is generated after the last slot of a receive frame ends when the ESSI is in the Network mode. The interrupt occurs regardless of the receive mask register setting.

Note: The RLIE bit is disabled when DC = 00.

12.8.9.15 Transmit Last Slot Interrupt Enable (TLIE)—Bit 0

- 0 = The Transmit last slot interrupt is disabled.
- 1 = An interrupt is generated at the beginning of the last slot of a transmit frame when the ESSI is in the Network mode. The interrupt occurs regardless of the transmit mask register setting.

Note: The TLIE bit is disabled when DC = 00.

12.8.10 ESSI Control Register 4 (SCR4)

The ESSI Control Register 4 (SCR4) is one of five, 16-bit, read/write control registers used to direct the operation of the ESSI. The SCR4 controls the configuration of the ESSI I/O pins and specifies the values of the Output Flag controls.

The Power-On Reset (POR) clears all of the SCR4 bits. The ESSI reset (ESSIEN = 0) does not affect the SCR4 bits. The SCR4 bits are discussed in the following paragraphs.

Base + $7	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read						TXSF0	TXSF1	TXSF2	0	SSC1	SCKD	SCD2	SCD1	SCD0	OF1	OF0
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 12-24. ESSI Control Register 4 (SCR4)

See Programmer's Sheet on Appendix page B - 76

12.8.10.1 Reserved—Bits 15–11

These bits are reserved or not implemented. They cannot be read nor modified by writing.

12.8.10.2 Transmit Status Flag Control 0 (TXSF0)—Bit 10

- 0 = The indicated transmitter does not affect the status bits.
- 1 = The indicated transmitter status will affect the TFF, TDE, TUE, and TFE status bits. The status from all selected transmitters is *ORed* to create the register status bit. These bits should be set to match the intended operation of the TE0, TE1, and TE2 control bits.

12.8.10.3 Transmit Status Flag Control 1 (TXSF1)—Bit 9

- 0 = The indicated transmitter does not affect the status bits.
- 1 = The indicated transmitter status will affect the TFF, TDE, TUE, and TFE status bits. The status from all selected transmitters is *ORed* to create the register status bit. These bits should be set to match the intended operation of the TE0, TE1, and TE2 control bits.

12.8.10.4 Transmit Status Flag Control 2 (TXSF2)—Bit 8

- 0 = The indicated transmitter does not affect the status bits.
- 1 = The indicated transmitter status will affect the TFF, TDE, TUE, and TFE status bits. The status from all selected transmitters is *ORed* to create the register status bit. These bits should be set to match the intended operation of the TE0, TE1, and TE2 control bits.

12.8.10.5 Reserved—Bit 7

This bit is reserved or not implemented. It is read as, and written with a 0.

12.8.10.6 Select SC1 (SSC1)—Bit 6

This bit controls the functionality of the SC1 signal. This control bit is used only if SYN = 1 and TE2 = 0.

- 0 = The SC1 acts as the serial I/O flag (IF1/OF1).
- 1 = When configured as an output (SCD1 = 1), the SC1 signal acts as the transmitter 0 driver-enable, enabling an external buffer for the transmitter 0 output.

Note: If SSC1 = 1 and SCD1 = 0, the configuration is invalid.

12.8.10.7 Clock Source Direction (SCKD)—Bit 5

This bit configures the source of the clock signal used to clock the Transmit Shift register in Asynchronous mode and both Transmit and Receive Shift registers are in Synchronous mode.

- 0 = SCK pin is configured as an input. This pin is used as the clock source.
- 1 = Internal transmit clock generator is selected. This clock is output on the SCK pin.

12.8.10.8 Serial Control Direction 2 (SCD2)—Bit 4

This bit controls the direction of the SC2 pin.

- 0 = SC2 is an input.
- 1 = SC2 is an output.

12.8.10.9 Serial Control Direction 1 (SCD1)—Bit 3

This bit controls the direction of the SC1 pin depending on the state of other controls as shown in **Table 12-15.**

- 0 = SC1 is an input
- 1 = SC1 is an output

Table 12-15. Control Modes Where SCD1 is Used

SYN	TE2	Functional State
0	x	SCD1 control direction of SC1
1	0	SC1 is used as flag 1 where SCD1 determines whether it is an output flag or an output flag
1	1	SC1 is an output

12.8.10.10 Serial Control Direction 0 (SCD0)—Bit 2

This bit controls the direction of the SC0 pin depending on the state of other controls as shown in **Table 12-16.**

- 0 = SC0 is an input
- 1 = SC0 is an output

Table 12-16. Control Modes Where SCD0 is Used

SYN	TE2	Functional State
0	x	SCD0 control direction of SCI0
1	0	SC0 is used as flag 0 where SCD0 determines whether it is an output flag or an output flag
1	1	SC0 is an output

12.8.10.11 Serial Output Flag 1 (OF1)—Bit 1

When SC1 is configured for Output Flag functionality, data provided in **Table 12-15,** this control bit determines the state of the output pin. Changes in the OF1 bit will appear on the SC1 pin at the beginning of the next frame in the Normal mode, or at the beginning of the next time slot in the Network mode.

12.8.10.12 Serial Output Flag 0 (OF0)—Bit 0

When SC0 is configured for Output Flag functionality, provided in **Table 12-16,** this control bit determines the state of the output pin. Changes in the OF0 bit will appear on the SC0 pin at the beginning of the next frame in the Normal mode, or at the beginning of the next time slot in the Network mode.

12.8.11 ESSI Transmit and Receive Control Registers (STXCR, SRXCR)

The STXCR and SRXCR are 16-bit, read/write control registers used to direct the operation of the ESSI. These registers control the ESSI clock generator bit and frame sync rates, word length, and number of words per frame for the serial data. Please refer to **Figure 12-35** to better understand how these fields are used in the clock generators.

The STXCR register is dedicated to the transmit section. The SRXCR register is dedicated to the receive section, except in the Synchronous mode where the STXCR register controls both the receive and transmit sections.

Power-On Reset (POR) clears all STXCR and SRXCR bits. The ESSI reset (ESSIEN = 0) does not affect the STXCR and SRXCR bits. The control bits are described in the ESSI Transmit Control Register (STXCR).

Base + $8	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	PSR	WL		DC					PM							
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 12-25. ESSI Transmit Control Register (STXCR)

See Programmer's Sheet on Appendix page B - 78

Base + $9	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	PSR	WL		DC					PM							
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 12-26. ESSI Receive Control Register (SRXCR)

See Programmer's Sheet on Appendix page B - 78

12.8.11.1 Prescaler Range (PSR)—Bit 15

This bit controls a fixed divide-by-eight prescaler in series with the variable prescaler. It extends the range of the prescaler for those cases where a slower bit clock is desired.

- 0 = When the PSR bit is cleared, the fixed prescaler is by-passed.
- 1 = When the PSR bit is set, the fixed divide-by eight prescaler is operational. This allows a 128kHz master clock to be generated for MC1440x series codecs. The maximum internally generated bit clock frequency is $f_{IP\_CLK}/(2\times2)$ and the minimum internally generated bit clock frequency is $f_{IP\_CLK}/(4\times2\times8\times256\times2)$.

12.8.11.2 Word Length Control (WL)—Bits 14–13

These bits are used to select the length of the data words being transferred by the ESSI. Word lengths of 8, 10, 12, or 16 bits can be selected. **Table 12-17** denotes the WL two-bit field encoding.

Table 12-17. WL Encoding

WL	Number of Bits/Word
00	8
01	10
10	12
11	16

These bits control the Word Length Divider shown in the ESSI Clock Generator. The WL control bit also controls the frame sync pulse length when the TFSL (or RFSL) bit is cleared.

12.8.11.3 Frame Rate Divider Control (DC)—Bits 12–8

This bit field controls the divide ratio for the programmable frame rate dividers. The divide ratio operates on the word clock. In the Normal mode, this ratio determines the word transfer rate. The divide ratio ranges from 1 to 32 (DC[4:0] = 00000 to 11111) in the Normal mode. A divide ratio of one (DC = 00000) provides continuous periodic data word transfer. A bit-length sync must be used in this case. In the Network mode, this ratio sets the number of words per frame. The divide ratio ranges from 2 to 32 (DC[4:0] = 00001 to 11111) in the Network mode. A divide ratio of one (DC = 00000) in the Network mode is a special case (on Demand mode) not supported.

12.8.11.4 Prescaler Modulus Select (PM)—Bits 7–0

This bit field specifies the divide ratio of the prescale divider in the ESSI clock generator. This prescaler is used only in Internal Clock mode to divide the internal peripheral clock. A divide ratio from 1 to 256 (PM = $00 to $FF) can be selected. The bit clock output is available at the SCK or SC0 clock pins. The bit clock on the ESSI can be calculated from the peripheral clock value using the following equation:

$f_{FIX\_CLK} = f_{IPBus\_CLK/4}$ if DIV4DIS = 0

$f_{FIX\_CLK} = f_{IPBus\_CLK}$ if DIV4DIS = 1

$f_{INT\_BIT\_CLK} = f_{FIX\_CLK} /[4 \times (7 \times PSR + 1) \times (PM + 1)]$

$f_{FRAME\_SYN\_CLK} = (f_{INT\_BIT\_CLK})/[(DC + 1) \times WL]$ where DC = DC and WL = 8, 10, 12, or 16

For example if an 8kHz sampling rate is desired, with 8-bit words operating in the Normal mode, the following parameters can be used:

$f_{IPBus\_CLK} = f_{SYSTEM\_CLK /2}$ = 120MHz / = 60mhz

$f_{FIX\_CLK} = f_{IPBus\_CLK}$ = 60MHz DIV4DIS = 1

$f_{INT\_BIT\_CLK} = f_{FIX\_CLK} /[4 \times (7 \times PSR + 1) \times (PM + 1)]$

= 60 MHz /[4 x 1 x 117] = 128.2kHz PS = 0, PM = 116

$f_{FRAME\_SYN\_CLK} = (f_{INT\_BIT\_CLK})/[(DC + 1) \times WL]$ DC = 1 = 128kHz/[2 x 8]

= 8.012kHz

The bit clock output is also available internally for use as the bit clock to shift the Transmit and Receive Shift registers. Careful choice of the crystal oscillator frequency and the prescaler modulus provides the telecommunication industry standard codec master clock frequencies of 2.048MHz and 1.536MHz to be generated; however, such applications will typically use external clocks and frame syncs. **Table 12-18** provides examples of PM, PSR, and DIV4DIS values available to be used to generate different bit clocks.

Table 12-18. Chip Clock Rates as a Function of ESSI Bit Clock Frequency and Prescale Modulus

Frame Rate	Words/Frame (DC+1)	Bits/ Word	Bit Clock Rate (Hz)	PM	PSR	FIX_CLK Rate	DIV4DIS	IP_CLK Rate[1]	SYS_CLK Rate[2]
8000	32	8	2,048,000	6	0	57,344,000	1	57,344,000	114,688,000
8000	32	8	2,048,000	7	0	65,536,000	1	65,536,000	131,072,000
8000	24	8	1,536,000	8	0	55,296,000	1	55,296,000	110,592,000
8000	24	8	1,536,000	9	0	61,440,000	1	61,440,000	125,880,000
8000	2	10	160,000	10	1	56,320,000	1	56,320,000	112,640,000
8000	2	10	160,000	11	1	61,440,000	1	61,440,000	122,880,000
8000	2	10	160,000	22	0	14,720,000	0	58,880,000	117,760,000
8000	2	10	160,000	92	0	59,520,000	1	59,520,000	119,040,000
8000	2	10	160,000	93	0	60,160,000	1	60,160,000	120,320,000
8000	2	12	192,000	8	1	55,296,000	1	55,296,000	110,592,000
8000	2	12	192,000	9	1	61,440,000	1	61,440,000	122,880,000
8000	2	12	192,000	18	0	14,593,000	0	58,368,000	116,736,000
8000	2	12	192,000	77	0	59,904,000	1	59,904,000	119,808,000
8000	2	12	192,000	78	0	60,672,000	1	60,672,000	120,344,000
8000	2	16	256,000	6	1	57,344,000	1	57,344,000	114,688,000
8000	2	16	256,000	7	1	65,536,000	1	65,536,000	131,072,000
8000	2	16	256,000	13	0	14,336,000	0	57,344,000	116,736,000
8000	2	16	256,000	57	0	59,904,000	1	59,904,000	118,784,000
8000	2	16	256,000	58	0	60,416,000	1	60,416,000	120,832,000

1. **Shaded cells denote** configuration where the required clock rate is higher than the chip specification can support.
2. IP_CLK is 1/2 the SYS_CLK rate.

12.8.12 Time Slot Register (STSR)

This register is used when data is not to be transmitted in an available transmit time slot, determined by the TSM registers. For the purposes of timing, the Time Slot register is a *write-only* register behaving like an alternate transmit data register, except instead of transmitting data, the STD pin signal is tri-stated. Using this register is important in avoiding overflow/underflow during inactive time slots.

Base + $A	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read																
Write	DUMMY REGISTER WRITTEN DURING INACTIVE TIME SLOTS (NETWORK MODE)															
Reset	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

Figure 12-27. ESSI Time Slot Register (STSR)

See Programmer's Sheet on Appendix page B - 79

12.8.13 ESSI FIFO Control/Status Register (SFCSR)

This register allows configuration of the Transmit and Receive FIFO Registers, and allows reporting of the amount of data contained in each FIFO

.

Base +$B	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	RFCNT				TFCNT				RFWM				TFWM			
Write																
Reset	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1

Figure 12-28. ESSI FIFO Control/Status Register (SFCSR)

See Programmer's Sheet on Appendix page B - 80

12.8.13.1 Receive FIFO Counter (RFCNT)—Bits 15–12

This *read-only* bit field indicates the number of data words in the RXFIFO. **Table 12-19** demonstrates the RFCNT bit field encoding.

Table 12-19. RFCNT Encoding

Bits	Description
0000	0 Data words in RXFIFO
0001	1 Data words in RXFIFO
0010	2 Data words in RXFIFO
0011	3 Data words in RXFIFO
0100	4 Data words in RXFIFO
0101	5 Data words in RXFIFO
0110	6 Data words in RXFIFO
0111	7 Data words in RXFIFO
1000	8 Data words in RXFIFO

12.8.13.2 Transmit FIFO Counter (TFCNT)—Bits 11–8

This *read-only* bit field indicates the number of data words in the TXFIFO. **Table 12-20** illustrates the TFCNT bit field encoding. The specific FIFO reported on is determined by the first TXSFn control bit enabled; that is to say, if TXSF0 is set, the number of data words in TXFIFO0 is reported. When more than one transmitter is enabled, the TF1ERR and TF2ERR status bits will indicate any counter mismatches, described in **Section 12.8.7.**

Note: While loading the Transmit Data registers the FIFO counts may temporarily not match so the count status bits should only be checked after a complete set of writes to the Transmit Data registers.

Table 12-20. TFCNT Encoding

Bits	Description
0000	0 Data words in TXFIFO
0001	1 Data words in TXFIFO
0010	2 Data words in TXFIFO
0011	3 Data words in TXFIFO
0100	4 Data words in TXFIFO
0101	5 Data words in TXFIFO
0110	6 Data words in TXFIFO
0111	7 Data words in TXFIFO
1000	8 Data words in TXFIFO

12.8.13.3 Receive FIFO Full Watermark (RFWM)—Bits 7–4

This bit field controls the threshold the Receive FIFO Full (RFF) flag will be set. RFF is set whenever the data level in the RXFIFO reaches the selected threshold. For example, if RFWM = 1, RFF will be set after the ESSI receives 2 data words (one in SRX and the other in RXFIFO). **Table 12-22** shows the status of RFF for all data levels of the RXFIFO

Table 12-21. RFWM Encoding

Bits	Description
0000	Reserved
0001	RFF set when at least 1data word has been written to the RXFIFO. Set when RXFIFO = 1, 2, 3, 4, 5, 6, 7, or 8 data words
0010	RFF set when 2 or more data words have been written to the RXFIFO. Set when RXFIFO = 2, 3, 4, 5, 6, 7, or 8 data words
0011	RFF set when 3 or more data words have been written to the RXFIFO. Set when RXFIFO = 3, 4, 5, 6, 7, or 8 data words
0100	RFF set when 4 or more data words have been written to the RXFIFO. Set when RXFIFO = 4, 5, 6, 7, or 8 data words
0101	RFF set when 5 or more data words have been written to the RXFIFO. Set when RXFIFO = 5, 6, 7, or 8 data words
0110	RFF set when 6 or more data words have been written to the RXFIFO. Set when RXFIFO = 6, 7, or 8 data words
0111	RFF set when 7 or more data words have been written to the RXFIFO. Set when RXFIFO = 7, or 8 data words
1000	RFF set when 8 or more data words have been written to the RXFIFO. Set when RXFIFO = 8 data words

Table 12-22. Status of Receive FIFO Full Flag

Receive FIFO Watermark (RFWM)	Number of Data in RXFIFO								
	0	1	2	3	4	5	6	7	8
1	0	1	1	1	1	1	1	1	1
2	0	0	1	1	1	1	1	1	1
3	0	0	0	1	1	1	1	1	1
4	0	0	0	0	1	1	1	1	1
5	0	0	0	0	0	1	1	1	1
6	0	0	0	0	0	0	1	1	1
7	0	0	0	0	0	0	0	1	1
8	0	0	0	0	0	0	0	0	1

12.8.13.4 Transmit FIFO Empty Watermark (TFWM)—Bits 3–0

This bit field controls the threshold where the Transmit FIFO Empty (TFE) flag is set. TFE is set whenever the data level in the TXFIFO falls below the selected threshold. **Table 12-23** shows the status of TFE for all data levels of the TXFIFO.

Table 12-23. TFWM Encoding

Bits	Description
0000	Reserved
0001	TFE set when there is 1empty slot in TXFIFO. (default) Transmit FIFO Empty is set when TXFIFO <= 7 data
0010	TFE set when there are 2 or more empty slots in TXFIFO. Transmit FIFO Empty is set when TXFIFO <= 6 data
0011	TFE set when there are 3 or more empty slots in TXFIFO. Transmit FIFO Empty is set when TXFIFO <= 5 data
0100	TFE set when there are 4 or more empty slots in TXFIFO. Transmit FIFO Empty is set when TXFIFO <= 4 data
0101	TFE set when there are 5 or more empty slots in TXFIFO. Transmit FIFO Empty is set when TXFIFO <= 3 data
0110	TFE set when there are 6 or more empty slots in TXFIFO. Transmit FIFO Empty is set when TXFIFO <= 2 data
0111	TFE set when there are 7 or more empty slots in TXFIFO. Transmit FIFO Empty is set when TXFIFO <= 1 data
1000	TFE set when there are 8 or more empty slots in TXFIFO. Transmit FIFO Empty is set when TXFIFO <= 0 data

Table 12-24. Status of Transmit FIFO Empty Flag

Transmit FIFO Watermark (TFWM)	Number of Data in TXFIFO								
	0	1	2	3	4	5	6	7	8
1	1	1	1	1	1	1	1	1	1
2	1	1	1	1	1	1	1	0	0
3	1	1	1	1	1	1	0	0	0
4	1	1	1	1	1	0	0	0	0
5	1	1	1	1	0	0	0	0	0
6	1	1	1	0	0	0	0	0	0
7	1	1	0	0	0	0	0	0	0
8	1	0	0	0	0	0	0	0	0

12.8.14 Transmit Slot Mask Registers (TSMA, TSMB)

The Transmit Slot Mask Registers are two 16-bit read/write registers. In the Network mode, these registers are used by the transmitter to determine which action to take in the current transmission slot. Depending on the setting of the bits, the transmitter either tri-states the transmitter data signal or transmits a data word and generates the appropriate transmit status. TSMA and TSMB can be viewed as a single 32-bit register named TSM. Bit *n* in TSM (TSMn) is an enable/disable control bit for transmission in slot *N*.

Base + $C	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	TSMA [15:0]															
Write																
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Figure 12-29. Transmit Slot Mask Register (TSMA)

See Programmer's Sheet on Appendix page B - 82

Base + $D	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	TSMB [31:16]															
Write																
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Figure 12-30. Transmit Slot Mask Register (TSMB)

See Programmer's Sheet on Appendix page B - 82

- 0 = The transmit data signal of the transmitter is tri-stated during transmit time slot *N*. Data is not transferred to the TXSR, therefore transmit status flags are not changed.

- 1 = The transmit sequence proceeds normally. Data is transferred from the STX register (or TXFIFO, if enabled) to the Shift register during slot *N*. Appropriate flags are set.

The DSC is interrupted only for enabled slots. Data written to the STX register when the Transmitter Empty (or Transmit FIFO Empty) Interrupt Request is being serviced is transmitted in the next enabled transmit time slot.

The TSM slot mask does not conflict with the STSR. Even if a slot is enabled in the TSM register, the user may choose to write to the STSR to tri-state the signals of the enabled transmitters during the next transmission slot. Setting the bits in the TSM register affects the next frame transmission. The frame currently being transmitted is not affected by the new TSM setting. If the TSM is read, it shows the current setting.

An ESSI reset (ESSIEN = 0) does not affect the contents of the TSM registers. After a hardware RESET signal, or executing a software RESET instruction, the TSM register is reset to $FFFFFFFF; that value enables all 32 slots for data transmission. The transmit DC setting determines how many of these control bits is actually used.

12.8.15 Receive Slot Mask Registers (RSMA, RSMB)

The Receive Slot Mask Registers are two 16-bit read/write registers. In the Network mode, these registers are used by the receiver to determine which action to take in the current time slot. Depending on the setting of the bits, the receiver either ignores the receiver data signal(s), or receives a data word, generating the appropriate receive status.

RSMA and RSMB can be viewed as one 32-bit register, RSM. Bit *n* in RSM (RSMn) is an enable/disable control bit for time slot *N*.

Base + $E	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	RSMA [15:0]															
Write																
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Figure 12-31. Receive Slot Mask Register (RSMA)

See Programmer's Sheet on Appendix page B - 83

Base + $F	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	RSMB [31:16]															
Write																
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Figure 12-32. Receive Slot Mask Register (RSMB)

See Programmer's Sheet on Appendix page B - 83

- 0 = Data is not transferred from the Receive Shift Register (RXSR) to the Receive Data (SRX) register, therefore the RDR and ROE flags are not set.
- 1 = The receive sequence proceeds normally. Data is received during slot *N*, and the RDR flag is set.

During a disabled slot, no receiver full interrupt is generated. The DSC is interrupted only for enabled slots.

When the bits in the RSM are changed, their settings affect the next frame reception. The frame currently being received is not affected by the new RSM setting. If the RSM is read, it shows the current setting. An ESSI reset (ESSIEN = 0) does not affect the contents of the TSM registers. After a hardware RESET signal or executing a software RESET instruction, the RSM register is reset to $FFFFFFFF; that value enables all 32-time slots for data reception. The receive DC setting determines how many of these control bits are actually used.

12.9 Clocks

The ESSI uses the following three clocks, illustrated in **Figure 12-33** and **Figure 12-34:**

- Bit clock–Used to serially clock the data bits in and out of the ESSI port
- Word clock–Used to count the number of data bits per word (8, 10, 12, or 16 bits)
- Frame clock–Used to count the number of words in a frame

The bit clock is used to serially clock the data. The clock is visible on the SCK, for Asynchronous transmit or Synchronous modes and SC0, for asynchronous receive clock operation pins. The word clock is an internal clock used to determine when transmission of an 8, 10, 12, or 16 bit word has completed. The word clock in turn then clocks the frame clock, which marks the beginning of each frame. The frame clock can be viewed on the SC2 and SC1 pins. The bit clock can be received from an ESSI clock pin or can be generated from the peripheral clock passed through a divider, illustrated in **Figure 12-35.**

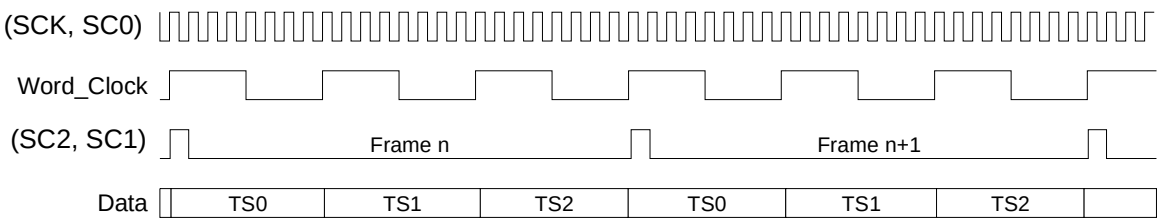


Figure 12-33. ESSI Clocking (8-Bit Words, 3 Time Slots / Frame)

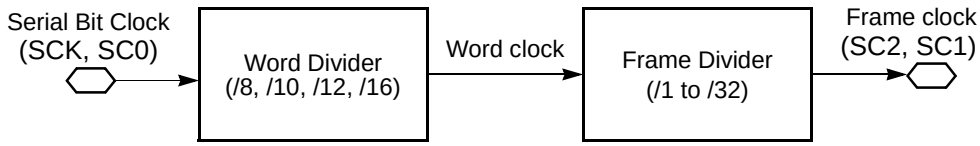


Figure 12-34. ESSI Clock Generation

Table 12-25. Clock Summary

Clock	Source	Characteristics
SCK	Internal/ External	Transmit data is changed on the rising edge of this clock. The RSCKP bit of the SCR2 register can invert the clock if required.
SC0	Internal/ External	Receive data is captured on the falling edge of this clock. The RSCKP bit of the SCR3 register can invert the clock if required.
SC2	Internal/ External	Transmit frames begin with the rising edge of this signal. See the definition of the TEFS bit of the SCR2 register for timing options. The TFSI bit can invert this signal if required.
SC1	Internal/ External	Receive frames begin with the rising edge of this signal. See the definition of the REFS bit of the SCR3 register for timing options.

12.10 Clock Operation Description

12.10.1 ESSI Clock and Frame Sync Generation

Data clock and frame sync signals can be generated internally by the ESSI or can be obtained from external sources. If internally generated, the ESSI clock generator is used to derive bit clock and frame sync signals from the peripheral clock. The ESSI clock generator consists of a selectable, fixed prescaler and a programmable prescaler for bit rate clock generation. A programmable frame rate divider and a word length divider are used for frame rate sync signal generation.

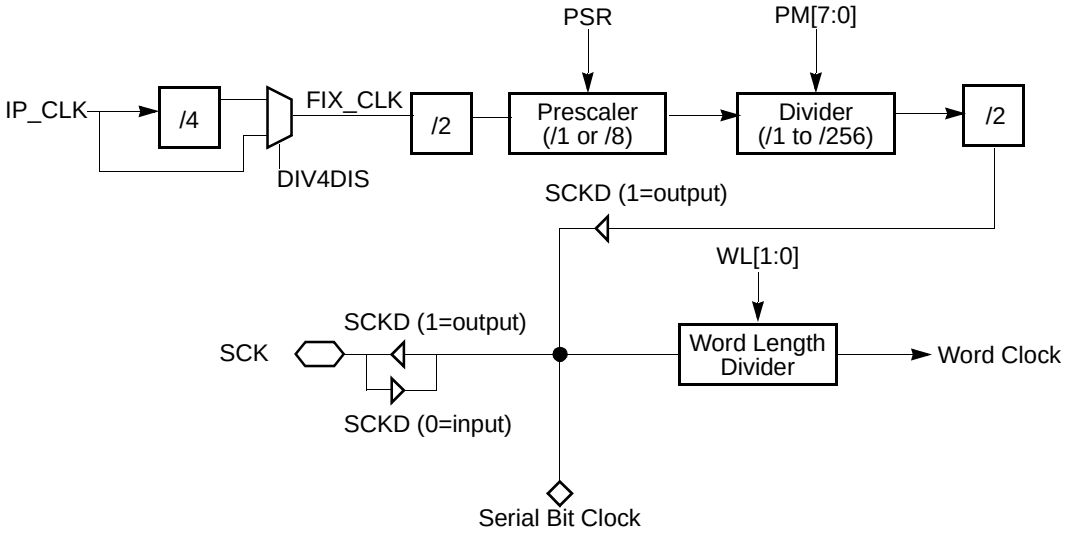


Figure 12-35. ESSI Transmit Clock Generator Block Diagram

Figure 12-36 demonstrates the frame sync generator block for the transmit section. When internally generated, both receive and transmit frame sync are generated from the word clock and are defined by the frame rate Divider (DC) field bits and the Word Length (WL) field bits of the ESSI Transmit Data Control Register (STXCR).

The receive section contains an equivalent circuit for its frame sync generator.

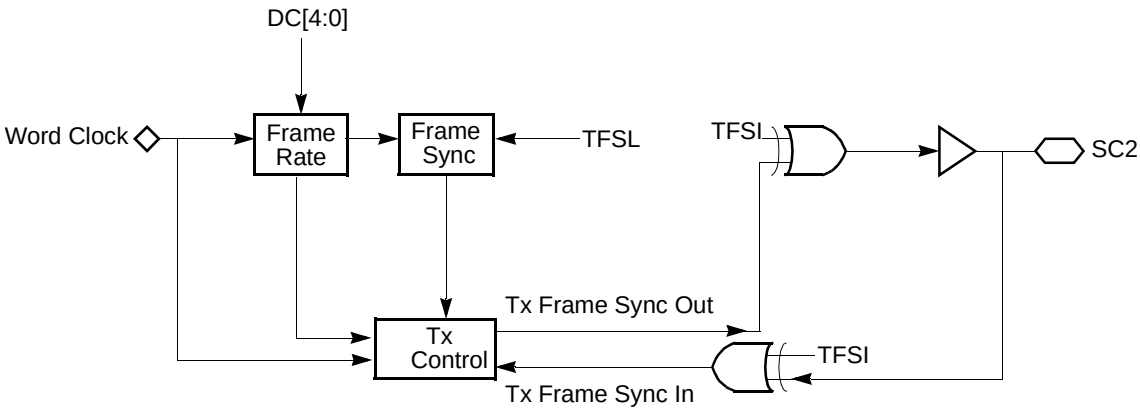


Figure 12-36. ESSI Transmit Frame Sync Generator Block Diagram

12.11 Resets

The ESSI is affected by Power-On Reset (POR) and ESSI reset.

POR is generated by asserting either the Reset pin or the (COP) timer reset. The POR initializes all control registers and clears the ESSIEN bit in SCR2 and disables the ESSI.

The ESSI reset is generated when the ESSIEN bit in the SCR2 is cleared. The ESSI status bits are preset to the same state produced by the POR. The ESSI control bits are unaffected. The ESSI reset is useful for selective reset of the ESSI without changing the present ESSI control bits and without affecting the other peripherals.

The correct sequence to initialize the ESSI is as follows:

1. Issue a Power-On or ESSI Reset.
2. Program the ESSI Control Registers.
3. Set the ESSIEN bit in SCR2.
4. Write data to the STX register(s).
5. Enable transmit and receive operations.

This procedure ensures proper operation of the ESSI by using the Power-On Reset (POR) or ESSI reset before changing any of the control bits listed in **Table 12-26.** These control bits should not be changed during ESSI operation.

Note: The ESSI bit clock must go low for at least one complete period to ensure proper ESSI reset.

Table 12-26. ESSI Control Bits Requiring Reset Before Change

Control Register	Bit
SRXCR STXCR	PSR WL[1:0] DC[4:0] PM[7:0]
SCR2	TEFS TFSL TFSI NET TSCKP TSHFD SYN

Table 12-26. ESSI Control Bits Requiring Reset Before Change (Continued)

Control Register	Bit
SCR3	TFEN RFEN REFS RFSL RFSI RSCKP RSHFD DIV4DIS
SCR4	SCD0 SCD1 SCD2 SDKD SSC1

12.12 Interrupts

The ESSI can generate up to six interrupt vectors, listed in **Table 12-27.**

12.13 Interrupt Operation Description

Table 12-27. Interrupt Summary

Interrupt	Source	Description
INTR + $0	Receiver	Receive data with exception
INTR + $2	Receiver	Receive data
INTR + $4	Receiver	Receive last slot interrupt. This interrupt may not be present in all implementations of the ESSI.
INTR + $6	Transmitter	Transmit data with exception
INTR + $8	Transmitter	Transmit data
INTR + $10	Transmitter	Transmit last slot interrupt. This interrupt may not be present in all implementations of the ESSI.

12.13.1 Receive Data With Exception

This interrupt can occur when receive interrupts are enabled via the RIE bit of the SCR2 register. When a data word is ready to transfer from the RXSR register to the SRX register and the previous SRX register data has not yet been read, the ROE bit is set and the exception interrupt will occur instead of the normal receive data interrupt. When the receive FIFO is enabled this interrupt will not occur until the RFF bit has been set, indicating the FIFO is full. The ROE bit is cleared when the SSR is read verifying the ROE bit is set before reading the SRX register data.

12.13.2 Receive Data (RX)

This interrupt can occur when receive interrupts are enabled via the RIE bit of the SCR2 register. When a data word is ready to transfer from the RXSR register to the SRX register, and the ROE bit is not set, an interrupt will occur indicating received data is available for processing. When the Receive FIFO is enabled, this interrupt will not occur until the Receive Watermark level of the FIFO is reached. If the FIFO is not enabled, an interrupt will occur for each data word received. This interrupt is cleared by reading the SSR, verifying the RDR bit is set before reading the SRX register data.

12.13.3 Receive Last Slot (RLS)

This interrupt occurs when the ESSI:

- Is in the Network mode
- Has been enabled via the RLIE bit of the SCR3 register
- The last slot of the frame has ended

The RLS bit is also set at this time. This interrupt is generated regardless of the Receive Mask register setting. This interrupt and the RLS bit are cleared by writing 1 to the RLS bit of the ESSI Status Register (SSR).

12.13.4 Transmit Data With Exception

This interrupt can occur when transmit interrupts are enabled via the TIE bit of the SCR2 register. When it is time to transfer data to the TXSR and data is not available in enabled STX or TXFIFO, the TUE status bit is set and the transmit data exception interrupt occurs. The TUE bit, and its interrupt, is cleared when the SSR is read then written to all the Transmit Data registers of the enabled transmitters or when written to the STSR.

12.13.5 Transmit Data (TX)

This interrupt can occur when transmit interrupts are enabled via the TIE bit of the SCR2 register. When data is transferred to the TXSR this interrupt will occur if more data is needed by any enabled transmitter.

If the transmit FIFO is not enabled this interrupt will occur for each data word transmitted. However, when the transmit FIFO is enabled, the interrupt will not occur until the Transmit Watermark level is reached. This interrupt is cleared by reading the SSR and writing data to the enabled STX registers. The interrupt may also be cleared by writing to the STSR.

12.13.6 Transmit Last Slot (TLS)

This interrupt occurs when the ESSI is in the Network mode at the beginning of the last slot of the transmit frame. The TLS bit is also set at this time. This exception occurs regardless of the Transmit Mask Register setting. This interrupt and the TLS bit are cleared by writing 1 to the TLS bit of the SSR.

12.14 User Notes

12.14.1 External Frame Sync Setup

When using external frame syncs, there must be at least four clocks after enabling the transmitter/receiver and before the first frame sync.

12.14.2 Maximum External Clock Rate

The maximum allowable rate for an external clock source is one fourth of the peripheral clock.

Chapter 13
Quad Timer (TMR)

13.1 Introduction

The Quad Timer (TMR) module contains four identical counter/timer groups. Each 16-bit counter/timer group contains a

- Prescaler
- Counter
- Load register
- Hold register
- Capture register
- Two Compare registers
- Two Status and Control registers

All except the prescaler are read/write registers.

Note: This document uses the terms *Timer* and *Counter* interchangeably because the counter/timers may perform either or both tasks.

The Load register provides the initialization value to the counter when the counter's terminal value has been reached. Hold registers capture the counter's value the instant any Counter register is read. This feature supports the reading of cascaded counters. The Capture register enables an external signal to take a *snapshot* of the counter's current value. The TMR_CMP1 and TMR_CMP2 registers provide the values to which the counter is compared. If a match occurs, the OFLAG signal can be set, cleared, or toggled. At match time, an interrupt is generated if enabled. The Prescaler provides different IPBus Clock time bases useful for clocking the counter. The Counter provides the ability to count internal or external events. Input pins are shared within a Timer module.

13.2 Features

The Quad TMR module design includes these distinctive capabilities:

- Four, 16-bit counters/timers
- Count up/down
- Counters are cascadable
- Count modulo can be programmed
- Maximum count rate equals peripheral clock for external clocks
- Maximum count rate equals peripheral clock for internal clocks
- Count once or repeatedly
- Counters can be preloaded

- Counters can share available input pins
- Separate prescaler for each counter
- Each counter has capture and compare capability

13.3 Operatng Modes

The TMR module design operates in only the Functional mode. Various counting modes are detailed in *Functional Description*, **Section 13.6.**

13.4 Block Diagram

Block diagram of the Quad TMR module is illustrated in **Figure 13-1.**

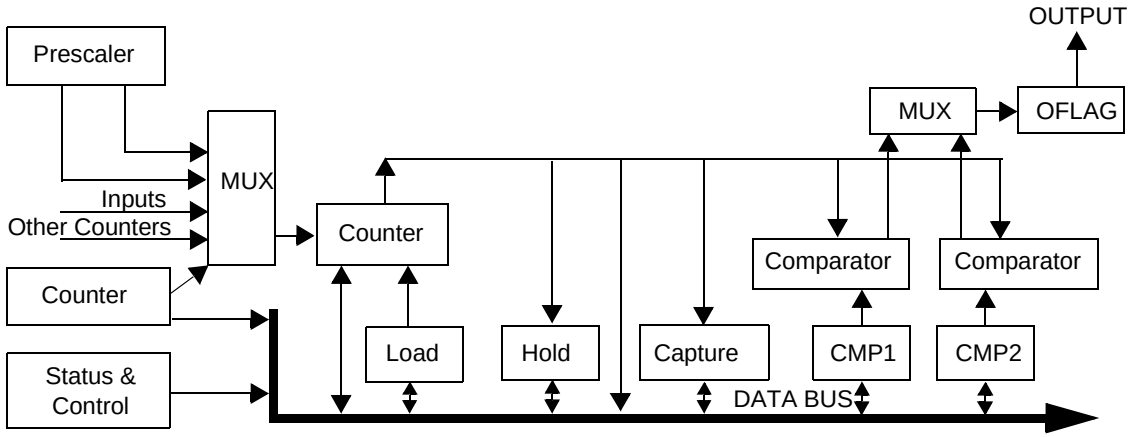


Figure 13-1. TMR Module Block Diagram

13.5 Signal Description

The TMR module has four external signals TIO[3:0] with the capability to be used as either inputs or outputs.

13.6 Functional Description

The counter/timer has two basic modes of operation:

1. Count internal or external events
2. Count an internal clock source while an external input signal is asserted, thus timing the width of the external input signal

The counter can count the rising, falling, or both edges of the selected input pin. The counter can decode and count quadrature encoded input signals. The counter can count up and down using dual inputs in a count with direction format. The counter's terminal count value (modulo) is programmable. The value loaded into the counter after reaching its terminal count is programmable. The counter can count repeatedly, or it can stop after completing one count cycle. The counter can be programmed to count to a programmed value and then immediately reinitialize, or it can count through the compare value until the count rolls over to zero.

The external inputs to each counter/timer can be shared among each of the four counter/timers within the module. The external inputs can be used as:

- Count commands
- Timer commands
- Trigger current counter value to be *captured*
- Generate interrupt requests

The polarity of the external inputs can be selected. For this implementation of the Timer (TMR), there are four input pins. The primary output of each timer/counter is the output signal, OFLAG. The OFLAG output signal can be set, cleared, or toggled when the counter reaches the programmed value. The OFLAG output signal may be output to an external pin shared with an external input signal (TIOx).

The OFLAG output signal enables each counter to generate square waves (PWM) or pulse stream outputs. The polarity of the OFLAG output signal is selectable.

Any counter/timer can be assigned as a Master (MSTR). A master's compare signal can be broadcasted to the other counter/timers within the module. The other counters can be configured to reinitialize their counters and/or force their OFLAG output signals to predetermined values when a Master's Counter/Timer compare event occurs.

13.7 Counting Modes Definitions

The selected external count signals are sampled at the TMR's base clock rate (60MHz) and then run through a transition detector. The maximum count rate is one-half of the base peripheral clock rate. Internal clock sources can be used to clock the counters at the peripheral clock rate.

If a counter is programmed to count to a specific value and then stop, the Count mode in the TMR_CTRL register is cleared when the count terminates.

13.7.1 Stop Mode

If the Count mode field is set to 000, the counter is inert. No counting will occur.

13.7.2 Count Mode

If the Count mode field is set to 001, the counter will count the rising edges of the selected clock source. This mode is useful for generating periodic interrupts for timing purposes, or counting external events such as *widgets* on a conveyor belt passing a sensor. If the selected input is inverted by setting the Input Polarity Select (IPS) bit, the negative edge of the selected external input signal is counted.

13.7.3 Edge Count Mode

If the Count mode field is set to 010, the counter will count both edges of the selected external clock source. This mode is useful for counting the changes in the external environment such as a simple encoder wheel.

13.7.4 Gated Count Mode

If the Count mode field is set to 011, the counter will count while the selected secondary input signal is high. This mode is used to time the duration of external events. If the selected input is inverted by setting the Input Polarity Select (IPS) bit, the counter will count while the selected secondary input is low.

13.7.5 Quadrature Count Mode

When the Count mode field is set to 100, the counter will decode the primary and secondary external inputs as quadrature encoded signals. Quadrature signals are usually generated by rotary or linear sensors used to monitor movement of motor shafts or mechanical equipment. The quadrature signals are square waves, 90 degrees out-of-phase. The decoding of quadrature signal provides both count and direction information. A timing diagram illustrating the basic operation of a quadrature incremental position encoder is provided in **Figure 13-2.**

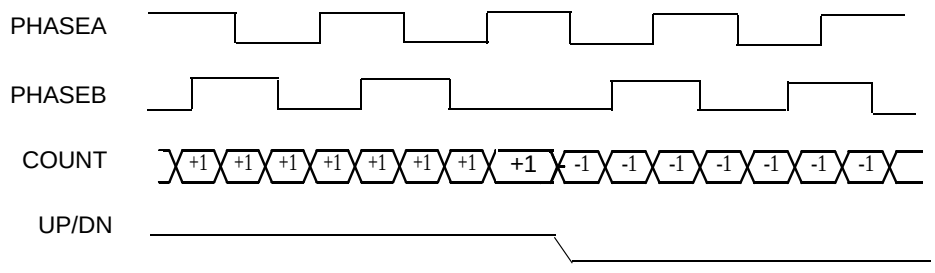


Figure 13-2. Quadrature Incremental Position Encoder

13.7.6 Signed Count Mode

If the Count mode field is set to 101, the counter counts the primary clock source while the selected secondary source provides the selected count direction (up/down).

13.7.7 Triggered Count Mode

If the Count mode field is set to 110, the counter will begin counting the primary clock source after a positive transition (Negative Edge if IPS = 1) of the secondary input occurs. The counting will continue until a compare event occurs, or another positive input transition is detected. If a second input transition occurs before a terminal count was reached, counting will stop. Subsequent odd numbered edges of the secondary input will restart counting, while even numbered edges will stop counting. This will continue until a compare event occurs.

13.7.8 One-Shot Mode

This is a sub mode of triggered event Count mode if the Count mode field is set to 110 while:

- Count Length (LENGTH) is set
- OFLAG Output mode is set to 101
- ONCE bit of the Control (CTRL) register is set to 1

In the above setting, the counter works in a One-Shot mode. An external event causes the counter to count. When terminal count is reached, the OFLAG output is asserted. This delayed output assertion can be used to provide timing delays.

13.7.9 Cascade Count Mode

If the Count mode field is set to 111, the counter's input is connected to the output of another selected counter. The counter will count up and down as compare events occur in the selected source counter. This Cascade or Daisy Chained mode enables multiple counters to be cascaded in order to yield longer counter lengths. When operating in the Cascade mode, a special high speed signal path is used not using the OFLAG Output signal. If the Selected Source Counter is counting up, and it experiences a compare event, the counter will be incremented. If the Selected Source Counter is counting down and it experiences a compare event, the counter will be decremented. Up to four counters may be cascaded to create a 64-bit wide synchronous counter. Whenever any counter is read within a Counter module, all of the counters' values within the module are captured in their respective Hold Registers. This action supports the reading of a cascaded counter chain. First read any counter of a cascaded counter chain, then read the Hold Registers of the other counters in the chain. The Cascaded Counter mode is synchronous.

Note: It is possible to connect counters together by using the other (non-cascade) Counter modes and selecting the outputs of other counters as a clock source. In this case, the

counters are operating in a *ripple* mode, where higher order counters will transition a clock later than a purely synchronous design.

13.7.10 Pulse Output Mode

The Counter will output a pulse stream of pulses with the same frequency of the selected clock source (can not be IPBus clock divided by one) if the counter is setup for:

- Count mode (mode = 001)
- The OFLAG Output mode is set to 111 (Gated Clock Output)
- The Count Once bit is set

The number of output pulses is equal to the compare value minus the initial value. This mode is useful for driving step motor systems.

Note: Primary count source must be set to one of the counter outputs for gated clock output mode.

13.7.11 Fixed Frequency PWM Mode

The Counter output yields a Pulse Width Modulated (PWM) signal with a frequency equal to the count clock frequency divided by 65,536. It has a pulse width duty cycle equal to the compare value divided by 65,536 if the counter is setup for:

- Count mode (mode = 001)
- Count through roll-over (Count Length = 0)
- Continuous count (Count Once = 0)
- OFLAG Output mode is 110 (set on compare, cleared on counter rollover)

This mode of operation is often used to drive PWM amplifiers used to power motors and inverters.

13.7.12 Variable Frequency PWM Mode

If the counter is setup for:

- Count mode (Mode = 001)
- Count till compare (Count Length = 1)
- Continuous count (Count Once = 0)
- OFLAG Output mode is 100 (toggle OFLAG and alternate compare registers)

the counter output yields a Pulse Width Modulated (PWM) signal whose frequency and pulse width is determined by the values programmed into the TMR_CMP1 and TMR_CMP2 registers,

and the input clock frequency. This method of PWM generation has the advantage of allowing almost any desired PWM frequency and/or constant on or off periods. This mode of operation is often used to drive PWM amplifiers used to power motors and inverters.

13.7.13 Compare Registers Use

The dual Compare registers (TMR_CMP1 and TMR_CMP2) provide a bidirectional modulo count capability. The CMP1 Register is used when the counter is counting *up*, and the CMP2 Register is used when the counter is counting *down*. The only exception is when the counter is operating with alternating compare registers. The CMP1 Register should be set to the desired maximum count value or $FFFF to indicate the maximum unsigned value prior to roll-over, and the CMP2 Register should be set to the maximum negative count value or $0000 indicating the maximum unsigned value prior to roll-under.

If the Output mode is set to 100, the OFLAG will toggle while using alternating Compare registers. In this Variable Frequency PWM mode, the CMP2 value defines the desired pulse width of the *on-time*, and the CMP1 Register defines the *off-time*. The Variable Frequency PWM mode is defined for positive counting only.

One must be careful when changing CMP1 and CMP2 while the counter is active. If the counter has already passed the new value, it will count to $FFFF or $0000, roll over/under, and then begin counting toward the new value. (The check is for Count = Cmpx, not Count> = Cmp1 or Count < = Cmp2).

13.7.14 Capture Register Use

The Capture Register stores a copy of the counter’s value when an input edge (positive, negative, or both) is detected. Once a capture event occurs, no further updating of the Capture Register will occur until the Input Edge Flag (IEF) is cleared by writing 0 to the IEF.

13.8 Module Memory Map

There are eight registers on the TMR peripheral described in **Table 13-1.**

Table 13-1. TMR Module Memory Map (TMR_BASE = $1FFE80)

Address Offset	Register Acronym	Register Name	Access Type	Chapter Location
Base + $0, $8, $10, $18	CMP1	Timer Channel Compare Register 1	Read/Write	Section 13.9.3
Base + $1, $9, $11, $19	CMP2	Timer Channel Compare Register 2	Read/Write	Section 13.9.4
Base + $2, $A, $12, $1A	CAP	Timer Channel Capture Register	Read/Write	Section 13.9.5
Base + $3, $B, $13, $1B	LOAD	Timer Channel Load Register	Read/Write	Section 13.9.6
Base + $4, $C, $14, $1C	HOLD	Timer Channel Hold Register	Read/Write	Section 13.9.7
Base + $5, $D, $15, $1D	CNTR	Timer Channel Counter Register	Read/Write	Section 13.9.8
Base + $6, $E, $16, $1E	CTRL	Timer Channel Control Register	Read/Write	Section 13.9.1
Base + $7, $F, $17, $F	SCR	Timer Channel Status/Control Reg.	Read/Write	Section 13.9.2

Add. Offset	Register Name		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
$0, $8, $10, $18	CMP1	R W	COMPARISON VALUE															
$1, $9, $11, $19	CMP2	R W	COMPARISON VALUE															
$2, $A, $12, $1A	CAP	R W	CAPTURE VALUE															
$3, $B, $13, $1B	LOAD	R W	LOAD VALUE															
$4, $C, $14, $1C	HOLD	R W	HOLD VALUE															
$5, $D, $15, $1D	CNTR	R W	COUNTER															
$6, $E, $16, $1E	CTL	R W	CM			PCS				SCS		ONCE	LENGTH	DIR	EXT INIT	OM (OFLAG)		
$7, $F, $17, $1F	SCR	R	TCF	TCFIE	TOF	TOFIE	IEF	IEFIE	IPS	INPUT	CAPTURE MODE		MSTR	EEOF	VAL		OPS	OEN
																FORCE		

R 0 Read as 0
W Reserved

Figure 13-3. TMR Register Map Summary

13.9 Register Descriptions (TMR_BASE = $1FFE80)

13.9.1 Timer Control Registers (CTL)

There are four Timer Control Registers in this occurrence. Their addresses are:

TMRA0_CTRL (Timer A, Channel 0 Control)—Address: TMRA_BASE + $6
TMRA1_CTRL (Timer A, Channel 1 Control)—Address: TMRA_BASE + $E
TMRA2_CTRL (Timer A, Channel 2 Control)—Address: TMRA_BASE + $16
TMRA3_CTRL (Timer A, Channel 3 Control)—Address: TMRA_BASE + $1E

Base + $6, $E, $16, $1E	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	CM			PCS				SCS		ONCE	LENGTH	DIR	EXT INIT	OM (OFLAG)		
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 13-4. TMR Control Register (CTL)

See Programmer's Sheet on Appendix page B - 90

13.9.1.1 Count Mode (CM)—Bits 15–13

These bits control the basic counting behavior of the counter.

- 000 = No operation
- 001 = Count rising edges of primary source[1]
- 010 = Count rising and falling edges of primary source
- 011 = Count rising edges of primary source while secondary input high active[1]
- 100 = Quadrature count mode, uses primary and secondary sources
- 101 = Count rising edges of primary source; secondary source specifies direction (1 = minus)[1]
- 110 = Edge of secondary source triggers primary count until compare
- 111 = Cascaded Counter mode (up/down)[2]

13.9.1.2 Primary Count Source (PCS)—Bits 12–9

These bits select the primary count source.

- 0000 = Counter 0 input pin (TIO0)
- 0001 = Counter 1 input pin (TIO1)
- 0010 = Counter 2 input pin (TIO2)

1. Rising edges counted only when IPS = 0. Falling edges counted when IPS = 1.
2. Primary count source must be set to one of the counter outputs.

- 0011 = Counter 3 input pin (TIO3)
- 0100 = Counter 0 output pin (OFLAG0)
- 0101 = Counter 1 output pin (OFLAG1)
- 0110 = Counter 2 output pin (OFLAG2)
- 0111 = Counter 3 output pin (OFLAG3)
- 1000 = Prescaler (IPBus clock divide-by 1)
- 1001 = Prescaler (IPBus clock divide-by 2)
- 1010 = Prescaler (IPBus clock divide-by 4)
- 1011 = Prescaler (IPBus clock divide-by 8)
- 1100 = Prescaler (IPBus clock divide-by 16)
- 1101 = Prescaler (IPBus clock divide-by 32)
- 1110 = Prescaler (IPBus clock divide-by 64)
- 1111 = Prescaler (IPBus clock divide-by 128)

Note: A timer selecting its own output for input is not a legal choice. The result is no counting.

13.9.1.3 Secondary Count Source (SCS)—Bits 8–7

These bits provide additional information, such as direction used for counting. They also define the source used by both the Capture mode bits and the Input Edge Flag in the Channel Status and Control register.

- 00 = Counter 0 input pin (TIO0)
- 01 = Counter 1 input pin (TIO1)
- 10 = Counter 2 input pin (TIO2)
- 11 = Counter 3 input pin (TIO3)

13.9.1.4 Count Once (ONCE)—Bit 6

This bit select continuous or one-shot counting mode.

- 0 = Count repeatedly
- 1 = Count till compare and then stop. If counting up, successful compare occurs when counter reaches CMP1 value. If counting down, successful compare occurs when counter reaches CMP2 value.

13.9.1.5 Count Length (LENGTH)—Bit 5

This bit determines whether the counter counts to the compare value and then reinitializes itself, or the counter continues counting past the compare value (binary roll-over).

- 0 = Roll-over
- 1 = Count till compare, then reinstalled. If counting up, successful compare occurs when counter reaches CMP1 value. If counting down, successful compare occurs when counter reaches CMP2 value.[1]

13.9.1.6 Count Direction (DIR)—Bit 4

This bit selects either the normal count-up direction, or the reverse down direction.

- 0 = Count Up
- 1 = Count Down

13.9.1.7 External Initialization (EXT INIT)—Bit 3

This bit enables another counter/timer within the same module to force the re-initialization of this counter/timer when the other counter has an active compare event.

- 0 = External counter/timers can not force a re-initialization of this counter/timer.
- 1 = External counter/timers may force a re-initialization of this counter/timer.

13.9.1.8 Output Mode (OM)—Bits 2–0

These bits determine the mode of operation for the OFLAG output signal.

- 000 = Asserted while counter is active
- 001 = Clear OFLAG output on successful compare
- 010 = Set OFLAG output on successful compare
- 011 = Toggle OFLAG output on successful compare
- 100 = Toggle OFLAG output using alternating compare registers
- 101 = Set on compare, cleared on secondary source input edge
- 110 = Set on compare, cleared on counter rollover
- 111 = Enable Gated Clock output while counter is active[2]

Note: Unexpected results may occur if the Output mode field is set to use alternating Compare registers (mode 100) and the Count Once bit is set.

1. When the Output mode 0×4 is used, alternating values of CMP1 and CMP2 are used to generate successful compares. For example, when the Output mode is 0×4, the counter counts until CMP1 value is reached, reinitializes, then counts until CMP2 value is reached, reinitializes, then counts until CMP1 value is reached, and so on.
2. Primary count source must be set to one of the counter outputs.

13.9.2 Timer Channel Status and Control Registers (SCR)

There are four Timer Status and Control Registers in this occurrence. Their addresses are:

TMRA0_SCR (Timer A, Channel 0 Status and Control)—Address: TMRA_BASE + $7
TMRA1_SCR (Timer A, Channel 1 Status and Control)—Address: TMRA_BASE + $F
TMRA2_SCR (Timer A, Channel 2 Status and Control)—Address: TMRA_BASE + $17
TMRA3_SCR (Timer A, Channel 3 Status and Control)—Address: TMRA_BASE + $1F

Base + $7, $F, $17, $1F	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	TCF	TCFIE	TOF	TOFIE	IEF	IEFIE	IPS	INPUT	CAPTURE MODE		MSTR	EEOF	VAL	0	OPS	OEN
Write														FORCE		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 13-5. TMR Status and Control Register (SCR)

See Programmer's Sheet on Appendix page B - 93

13.9.2.1 Timer Compare Flag (TCF)—Bit 15

This bit is set when a successful compare occurs. Clear the bit by writing 0 to it.

13.9.2.2 Timer Compare Flag Interrupt Enable (TCFIE)—Bit 14

When set, this bit enables interrupts when the TCF bit is set.

13.9.2.3 Timer Overflow Flag (TOF)—Bit 13

This bit is set when the counter rolls over its maximum value $FFFF or $0000, depending on count direction. Clear the bit by writing 0 to it.

13.9.2.4 Timer Overflow Flag Interrupt Enable (TOFIE)—Bit 12

When set, this bit enables interrupts when the TOF bit is set.

13.9.2.5 Input Edge Flag (IEF)—Bit 11

This bit is set when a positive input transition occurs while the counter is enabled. Clear the bit by writing 0 to it.

Note: Setting the Input Polarity Select (IPS) bit enables the detection of negative input edge transitions detection. Also, the control register's secondary count source determines which external input pin is monitored by the detection circuitry.

13.9.2.6 Input Edge Flag Interrupt Enable (IEFIE)—Bit 10

When set, this bit enables interrupts when the IEF bit is set

13.9.2.7 Input Polarity Select (IPS)—Bit 9

When set, this bit inverts the input signal polarity.

13.9.2.8 External Input Signal (INPUT)—Bit 8

This bit reflects the current state of the external input pin selected via the secondary count source after application of the IPS bit. This is a *read-only* bit.

13.9.2.9 Input Capture Mode (Capture Mode)—Bits 7–6

These bits specify the operation of the Capture Register as well as the operation of the input edge flag.

- 00 = Capture function is disabled
- 01 = Load Capture Register on rising edge of input
- 10 = Load Capture Register on falling edge of input
- 11 = Load Capture Register on any edge of input

13.9.2.10 Master Mode (MSTR)—Bit 5

When set, this bit enables the Compare function's output to be broadcasted to the other counter/timers in the module. This signal then can be used to reinitialize the other counters and/or force their OFLAG signal outputs.

13.9.2.11 Enable External OFLAG Force (EEOF)—Bit 4

When set, this bit enables the compare from another counter/timer within the same module to force the state of this counters' OFLAG Output signal.

13.9.2.12 Forced OFLAG Value (VAL)—Bit 3

This bit determines the value of the OFLAG Output signal when a software triggered FORCE command occurs.

13.9.2.13 Force OFLAG Output (FORCE)—Bit 2

This *write-only* bit forces the current value of the VAL bit to be written to the OFLAG Output. Always read this bit as 0. The VAL and FORCE bits can be written simultaneously in a single write operation. Write to the FORCE bit only if the counter is disabled.

- 0 = No action
- 1 = Forces the current value of the VAL bit to be written to OFLAG Output

Note: Setting this bit while the counter is enabled may yield unpredictable results.

13.9.2.14 Output Polarity Select (OPS)—Bit 1

This bit determines the polarity of the OFLAG Output signal.

- 0 = True polarity
- 1 = Inverted polarity

13.9.2.15 Output Enable (OEN)—Bit 0

When set, this bit enables the OFLAG Output signal to be put on the external pin. Additionally, setting this bit connects a timer's output pin to its input. The polarity of the signal will be determined by the OPS bit.

13.9.3 Timer Channel Compare Register 1 (CMP1)

These read/write registers store the value used for comparison with counter value. There are four Timer Channel Compare Registers in this occurrence. Their addresses are:

TMRA0_CMP1 (Timer A, Channel 0 Compare 1)—Address:TMRA_BASE + $0
TMRA1_CMP1 (Timer A, Channel 1 Compare 1)—Address: TMRA_BASE + $8
TMRA2_CMP1 (Timer A, Channel 2 Compare 1)—Address:TMRA_BASE + $10
TMRA3_CMP1 (Timer A, Channel 3 Compare 1)—Address: TMRA_BASE + $18

Base + $0. $8, $10, $18	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	COMPARISON VALUE 1															
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 13-6. TMR Compare Register 1 (CMP1)

See Programmer's Sheet on Appendix page B - 84

13.9.4 Timer Channel Compare Register 2 (CMP2)

These read/write registers store the value used for comparison with counter value. There are four Timer Compare Registers in this occurrence. Their addresses are:

TMRA0_CMP2 (Timer A, Channel 0 Compare 2)—Address: TMRA_BASE + $1
TMRA1_CMP2 (Timer A, Channel 1 Compare 2)—Address: TMRA_BASE + $9
TMRA2_CMP2 (Timer A, Channel 2 Compare 2)—Address: TMRA_BASE + $11
TMRA3_CMP2 (Timer A, Channel 3 Compare 2)—Address: TMRA_BASE + $19

Base + $1, $9, $11, $19	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	COMPARISON VALUE 2															
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 13-7. TMR Compare Register 2 (CMP2)

See Programmer's Sheet on Appendix page B - 85

13.9.5 Timer Channel Capture Register (CAP)

These read/write registers store the values captured from the counters. There are four Timer Channel Hold Registers in this occurrence. Their addresses are:

TMRA0_CAP (Timer A, Channel 0 Capture)—Address: TMRA_BASE + $2
TMRA1_CAP (Timer A, Channel 1 Capture)—Address: TMRA_BASE + $A
TMRA2_CAP (Timer A, Channel 2 Capture)—Address: TMRA_BASE + $12
TMRA3_CAP (Timer A, Channel 3 Capture)—Address: TMRA_BASE + $1A

Base + $2, $A, $12, $1A	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	CAPTURE VALUE															
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 13-8. TMR Capture Register (CAP)

See Programmer's Sheet on Appendix page B - 86

13.9.6 Timer Channel Load Register (LOAD)

These read/write registers store the value used to load the counter. There are four Timer Channel Load Registers in this occurrence. Their addresses are:

TMRA0_LOAD (Timer A, Channel 0 Load)—Address: TMRA_BASE + $3
TMRA1_LOAD (Timer A, Channel 1 Load)—Address: TMRA_BASE + $B
TMRA2_LOAD (Timer A, Channel 2 Load)—Address: TMRA_BASE + $13
TMRA3_LOAD (Timer A, Channel 3 Load)—Address: TMRA_BASE + $1B

Base + $3, $B, $13, $1B	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	LOAD VALUE															
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 13-9. TMR Load Register (LOAD)

See Programmer's Sheet on Appendix page B - 87

13.9.7 Timer Channel Hold Register (HOLD)

These read/write registers store the channel's value whenever any counter is read. There are four Timer Channel Hold Registers in this occurrence. Their addresses are:

TMRA0_HOLD (Timer A, Channel 0 Load)—Address: TMRA_BASE + $4
TMRA1_HOLD (Timer A, Channel 1 Load)—Address: TMRA_BASE + $C
TMRA2_HOLD (Timer A, Channel 2 Load)—Address: TMRA_BASE + $14
TMRA3_HOLD (Timer A, Channel 3 Load)—Address: TMRA_BASE + $1C

Base + $4, $C, $14, $1C	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	HOLD VALUE															
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 13-10. TMR Hold Register (HOLD)

See Programmer's Sheet on Appendix page B - 88

13.9.8 Timer Channel Counter Register (CNTR)

There are four read/write Timer Channel Counter Registers in this occurrence. Their addresses are:

TMRA0_CNTR (Timer A, Channel 0 Counter)—Address: TMRA_BASE + $5
TMRA1_CNTR (Timer A, Channel 1 Counter)—Address: TMRA_BASE + $D
TMRA2_CNTR (Timer A, Channel 2 Counter)—Address: TMRA_BASE + $15
TMRA3_CNTR (Timer A, Channel 3 Counter)—Address: TMRA_BASE + $1D

Base + $5. $D, $15, $1D	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	COUNTER															
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 13-11. TMR Counter Register (CNTR)

See Programmer's Sheet on Appendix page B - 89

13.10 Resets

The TMR module can only be reset by the $\overline{\text{RST}}$ signal. This forces all registers to their reset state and clears the OFLAG signal if it is asserted. The counter will be turned off until the settings in the Control register are changed.

13.11 Interrupts

The TMR module can generate 12 interrupts, three for each of the four counters/channels.

13.11.1 Timer Compare Interrupts

These interrupts are generated when a successful compare occurs between a counter and it's compare registers while the Timer Compare Flag Interrupt Enable (TCFIE) is set in the TMR_SCR. These interrupts are cleared by writing 0 to the TCF bit in the appropriate TMR_SCR.

13.11.2 Timer Overflow Interrupts

These interrupts are generated when a counter rolls over its maximum value while the TCFIE bit is set in the TMR_SCR. These interrupts are cleared by writing 0 to the Timer Overflow Flag (TOF) bit of the appropriate TMR_SCR.

13.11.3 Timer Input Edge Interrupts

These interrupts are generated by a transition of the input signal (either positive or negative depending on IPS setting) while the Input Edge Flag Interrupt Enable (IEFIE) bit is set in the TMR_SCR. These interrupts are cleared by writing 0 to the IEF bit of the appropriate TMR_SCR.

Chapter 14
Time-Of-Day (TOD)

14.1 Introduction

The 5685x devices in this manual have a Time-of-Day (TOD) feature implemented as a sequence of counters to track elapsed time. The hardware is capable of tracking time up to 179.5 years, or 65,535 days.

TOD is comprised of a series of counters tracking elapsed seconds, minutes, hours, and days. The module requires an input clock ranging from 1-65536Hz integer values only. The clock is further scaled down to generate a 1Hz clock driving all of the time counters.

All of the time counters are loaded with the time-of-day upon enabling TOD. Time counters subsequently track the elapsed time. If required, the TOD feature can issue an alarm whenever selected current time registers match the time programmed into the enabled TOD alarm registers. The TOD can also provide an interrupt every second.

An appropriate external clock frequency must be provided and the CGM module must be configured so TOD clock input in integer HZ from 1-65536Hz is provided. Once completed, the internal clock prescaler within the TOD module can be configured to provide a 1Hz time-base clock to the module.

The CGM provides a bit to select one of two external clock prescalers. The oscillator TOD clock prescaler is a fixed／ 128 divider. The CGM TOD clock prescaler is a programmable divider followed by a fixed／ 2 divider. The programmable portion ranges from divide by 1 up to 2 to the 12th power.

The oscillator prescaler is provided for use with a low frequency (2 - 4MHz) external clock and it is provided for use in conjunction with the PLL. The CGM clock prescaler consumes more power and it is larger for use with direct external clock frequencies from 4MHz up to 240MHz.

It is preferable to use the highest frequency TOD clock input possible therefore to do as much clock division as possible using the internal prescaler within the TOD module. TOD clock frequencies below 10Hz are not recommended.

14.2 Features

- Separate counters for seconds, minutes, hours, and days with a 16-bit day capacity
- Separate read/write registers for seconds, minutes, hours, and days
- Alarm clock registers for seconds, minutes, hours, and days
- Alarm interrupt with independent enables for compare of seconds, minutes, hours and days
- 1Hz interrupt with enable
- TOD reset only at power-on, unaffected by reset pin, software reset, or COP reset
- Flexible clock prescaling for use with either external clocking or crystal oscillator
- Works with crystal frequency of 2 - 4MHz
- Capability to generate interrupt, pulling the part out of sleep
- Can be configured to generate an alarm at a designated time

14.3 Block Diagram

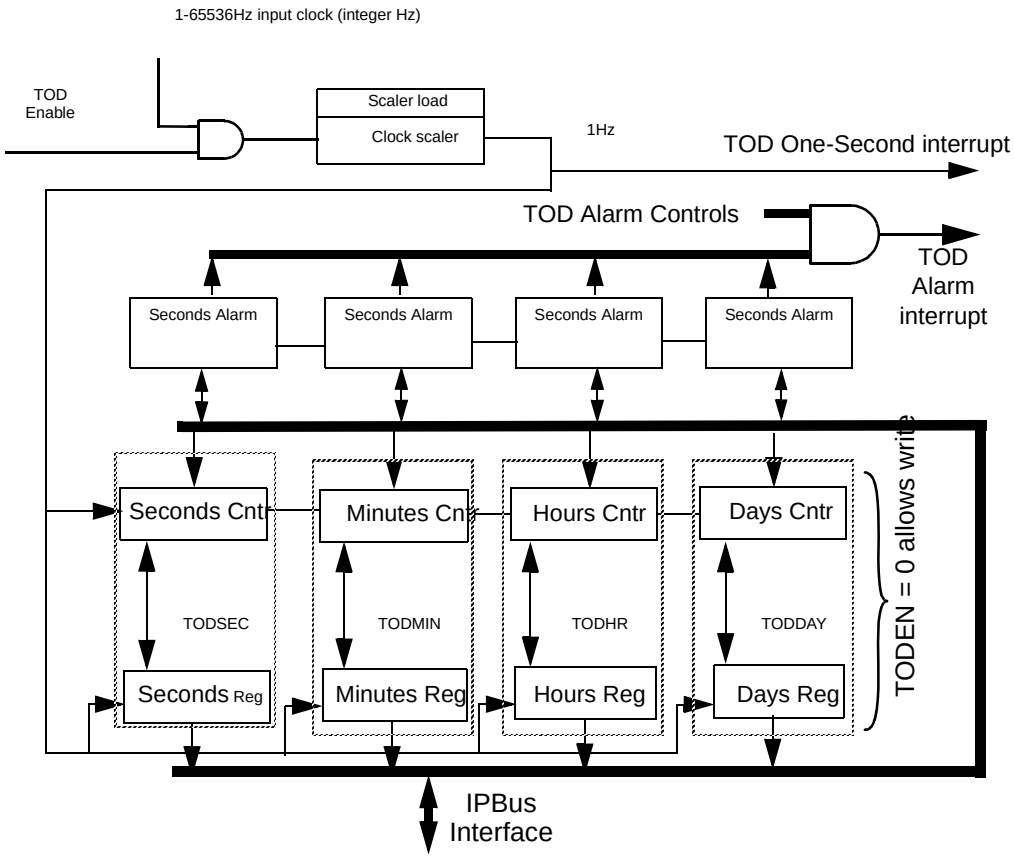


Figure 14-1. Time-of-Day Counter Operation Block Diagram

14.4 Module Memory Map

The base address of the TOD module is $1FFFC0.

Table 14-1. TOD Module Memory Map (TOD_BASE = $1FFFC0)

Address Offset	Register Acronym	Register Name	Access Type	Chapter Location
Base + $0	TODCS	Control Status Register	Read/Write	Section 14.6.2
Base + $1	TODCSL	Clock Scaler Register	*Read-Only*	Section 14.6.3
Base + $2	TODSEC	Seconds Register	*Read-Only*	Section 14.6.4
Base + $3	TODSAL	Seonds Alarm Register	*Read-Only*	Section 14.6.5
Base + $4	TODMIN	Minutes Register	*Read-Only*	Section 14.6.6
Base + $5	TODMAL	Minutes Alarm Register	*Read-Only*	Section 14.6.7
Base + $6	TODHR	Hours Register	*Read-Only*	Section 14.6.8
Base + $7	TODHAL	Hours Alarm Register	*Read-Only*	Section 14.6.9
Base + $8	TODDAY	Days Register	*Read-Only*	Section 14.6.10
Base + $9	TODDAL	Hours Alarm Register	*Read-Only*	Section 14.6.11

Add. Offset	Register Name		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
$0	TODCS	R	TODSIO	TODAL					TEST		TODDA	TODHA	TODMA	TODSA	TODSEN	TODAEN	TOD_LOCK	TODEN
		W																
$1	TODCSL	R	TIME-OF-DAY CLOCK SCALER															
		W																
$2	TODSEC	R											TODSEC					
		W																
$3	TODSAL	R											TODSAL					
		W																
$4	TODMIN	R											TODMIN					
		W																
$5	TODMAL	R											TODMAL					
		W																
$6	TODHR	R											TODHR					
		W																
$7	TODHAL	R												TODHAL				
		W																
$8	TODDAY	R	TODDAY															
		W																
$9	TODDAL	R	TODDAL															
		W																

R 0 Read as 0
W Reserved

Figure 14-2. TOD Register Map Summary

14.5 Functional Description

A sub block of TOD implements the scaler, seconds, minutes, hours and days counters. The counters can be initialized with the current time when the TOD module is disabled (when TODEN = 0).

14.5.1 Scaler

Depending upon the frequency of the clock fed to the TOD module, the appropriate value should be written to the clock scaler load register to derive a 1Hz clock. The scaler counter is a 16-bit up counter, counting from zero while less than the value specified in the clock scaler register. In normal operation, the scaler counter increments with TOD_CLK input (frequency range from 1-65536 Hz) if TODEN = 1. The base (1Hz) clock is generated when the scaler counter wraps and is used to clock the time-of-day counters and 1Hz interrupt.

14.5.2 Time Units

Time units (seconds, minutes, hours, days) are each implemented as counter/register pairs.

14.5.2.1 Time Counters

The time counters are a set of modulo counters that are clocked by the 1 Hz clock to maintain the current time in seconds, minutes, hours and days. They count only when TOD EN = 1. These counters maintain the time to the accuracy of the 1 Hz clock. They initialize to the current value in the time registers at the first 1Hz clock (one second) after TODEN is set to 1 and advance each second thereafter.

14.5.2.2 Time Registers

Time registers are provided to set and observe the corresponding counters. When TODEN is set to 1, the registers capture the counter values two IP_CLK cycles after the 1Hz clock and are readable but not writeable. When TODEN is set to 0, they will contain the current value of the time counters and they may be read or written.

14.5.2.3 Time-of-Day Lock Bit

When set, the TOD Lock Bit disables the continuous transfer of time-of-day counter values to the registers when TODEN is 1. This permits a read of each of the four time registers without the possibility of an in-process advancement of time values. The TOD_LOCK bit should remain set at 0 except while reading the time registers when TODEN = 1.

14.5.3 Stop Mode

During Stop Mode the TOD module will continue to operate normally. The TOD interrupts will still activate and can wake the processor up for TOD related processing.

14.5.4 General Information

The TOD alarms sub block is used to generate TOD IRQ signals. The TOD alarm registers TODSAL, TODMAL, TODHAL, and TODDAL, along with TOD Control/Status register (TODCS), are implemented in this sub block.

TOD generates two interrupt signals:

1. TOD Alarm Interrupt
2. TOD Second Interrupt

Both interrupt signals can wake up the device from its Stop mode.

14.5.5 Alarm Interrupt Flag and Outputs

The alarm interrupt feature provides a flexible means of detecting events based on time-of-day. The TOD module contains alarm registers for seconds, minutes, hours, and days. The control and status register contains individual enables for each of these registers (TODSA, TODMA, TODHA, TODDA) as well as an overall Alarm Interrupt Enable (TODAEN). An alarm interrupt triggers during the time-of-day counting process. This interrupt occurs when at least one of the alarm enables is set and the TODAEN is set.

All alarm registers can be enabled to generate an alarm at a designated time. All enabled alarm registers must match the corresponding time registers before the TOD alarm interrupt IRQ is generated. For example, if the minutes and hour alarms are enabled, then both the minutes and hours registers must match the respective alarm registers before the IRQ will occur.

The alarm interrupt triggers one TOD input clock period after the TOD_CLK edge, advancing the time-of-day counters to their activation value. To minimize latency, it is preferable to generate a faster TOD input clock and use a correspondingly higher value in the TOD module's clock scaler register. The alarm interrupt will not trigger until TODEN is set to one and all enabled alarm registers match the time-of-day counters. The interrupt will function in the Stop mode; however, the TOD module must be configured and enabled prior to entering the Stop mode.

Every time the alarm interrupt triggers the TOD Alarm interrupt occurred flag (TODAL) in the Control Status register is set. The TODAL is cleared by writing 0 to it while it contains 1. The interrupt will not be taken unless it is also enabled in the interrupt controller module.

14.5.6 1-Second Interrupt Flag and Outputs

The one-second interrupt feature provides a means of detecting the passage of one-second time intervals. The one-second interrupt is triggered and the TOD Seconds Interrupt (TODSID) occurred flag in the Status Control register is set on the rising edge of the 1Hz clock. The 1Hz clock is derived out of the scaler counter. This clock only operates while the TOD Enable (TODEN) is set. The TODSID bit is cleared by writing 0 to it while it contains a one. This interrupt will function in the Stop mode; however, the TOD module must be configured and enabled prior to entering the Stop mode.

The one-second interrupt output is used to signal a one-second interrupt to the interrupt controller. The interrupt will not be taken unless it is also enabled in the Interrupt Controller module.

14.6 Register Description (TOD_BASE = $1FFFC0)

The address of a register is the sum of a base address and an address offset. The base address is defined at the MCU level and the address offset is defined at the module level. The base address given for each register will be TOD_BASE.

14.6.1 TOD Register Map

Table 14-2. TOD Register Map

Register Address	Register Name	Register Function	Decimal Range	When Write Enabled	POR Value
$1FFFC0	TODCS	TOD Control/Status	N/C	Always	0
$1FFFC1	TODCSL	Clock Scaler Load	0-65535	TODEN = 0	0
$1FFFC2	TODSEC	Seconds	0-59	TODEN = 0	0
$1FFFC3	TODSAL	Seconds Alarm	0-59	Always	0
$1FFFC4	TODMIN	Minutes	0-59	TODEN = 0	0
$1FFFC5	TODMAL	Minutes Alarm	0-59	Always	0
$1FFFC6	TODHR	Hours	0-23	TODEN = 0	0
$1FFFC7	TODHAL	Hours Alarm	0-23	Always	0
$1FFFC8	TODDAY	Days	0-65535	TODEN = 0	0
$1FFFC9	TODDAL	Days Alarm	0-65535	Always	0

14.6.2 Time-of-Day Control Status (TODCS)

The TOD Control Status register controls TOD operation.

Base + $0	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	TODSIO	TODAL					TEST		TODDA	TODHA	TODMA	TODSA	TODSEN	TODAEN	TOD_LOCK	TODEN
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 14-3. TOD Control Status (TODCS)

See Programmer's Sheet on Appendix page B - 95

14.6.2.1 Time-of-Day 1-Second Interrupt Occurred Flag (TODSIO)—Bit 15

This bit is set when one-second interrupt occurs. This bit is cleared by writing 0 to the bit position. It must be cleared before exiting the Interrupt Service Routine (ISR).

14.6.2.2 Time-of-Day Alarm Interrupt Occurred Flag (TODAL)—Bit 14

This bit is set when TOD Alarm Interrupt occurs. This bit is cleared by writing 0 to the bit position. This bit must be cleared before exiting the Interrupt Service Routine.

14.6.2.3 Reserved—Bits 13–10

These bits are reserved or not implemented. They cannot be read nor modified by writing.

14.6.2.4 Test (TEST)—Bits 8–9

This bit field is reserved as factory test bits and must be written as 0.

14.6.2.5 Time-of-Day Days Alarm Enable (TODDA)—Bit 7

- 0 = Alarm interrupt ignores days counter
- 1 = Alarm interrupt requires match of days alarm register to days counter

14.6.2.6 Time-of-Day Hours Alarm Enable (TODHA)—Bit 6

- 0 = Alarm interrupt ignores hours counter
- 1 = Alarm interrupt requires match of hours alarm register to hours counter

14.6.2.7 Time-of-Day Minutes Alarm Enable (TODMA)—Bit 5

- 0 = Alarm interrupt ignores minutes counter
- 1 = Alarm interrupt requires match of minutes alarm register to minutes counter

14.6.2.8 Time-of-Day Seconds Alarm Enable (TODSA)—Bit 4

- 0 = Alarm interrupt ignores seconds counter
- 1 = Alarm interrupt requires match of seconds alarm register to seconds counter

14.6.2.9 Time-of-Day Seconds Interrupt Enable (TODSEN)—Bit 3

- 0 = Disables TOD seconds interrupt
- 1 = Enables TOD seconds interrupt

14.6.2.10 Time-of-Day Alarm Interrupt Enable (TODAEN)—Bit 2

- 0 = Disables TOD alarm interrupt
- 1 = Enables TOD alarm interrupt

14.6.2.11 Time-of-Day Lock (TOD_LOCK)—Bit 1

When set, this bit prevents the time registers from automatically updating to the current value of the time counters while the TOD module is enabled. It is used to avoid register updates while in the process of reading out the four time registers. It should be set only when in process of reading the current time while the TOD module is enabled.

- 0 = TOD registers not locked
- 1 = TOD registers locked

14.6.2.12 Time-of-Day Enable (TODEN)—Bit 0

Writing zero to TODEN disables the time-of-day counting; the time registers will contain the final value of the time counters and the registers become writable. When this field is written with 1, the seconds, minutes, hours, and days counters are loaded from the corresponding register and proceed to count in a normal time-of-day basis with seconds overflow incrementing minutes, minutes overflow incrementing hours, and so on through days overflow wrapping back to zero.

While time-of-day counting is enabled, the seconds, minutes, hours, and day registers update with a two IPBus clock delay to the value of the corresponding counter, so reading the registers returns the current time-of-day. In this mode, writing seconds register, minutes register, hour register and days register, or clock scaler is disabled.

14.6.3 Time-of-Day Clock Scaler (TODCSL)

Base + $1	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	TIME-OF-DAY CLOCK SCALER															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 14-4. TOD Clock Scaler (TODCSL)

See Programmer's Sheet on Appendix page B - 96

14.6.3.1 Time-of-Day Clock Scaler (TODCSL)—Bits 15-0

Setting this field to X divides the TOD Input Clock frequency by X + 1 to produce the time-base clock used for incrementing the counters. The clock generation system and this scaler must be configured so the time-base clock is precisely 1Hz for correct operation of the TOD module as the seconds counter increments with each time-base clock.

This register is always readable. It accepts writing only when TODEN = 0.

14.6.4 Time-of-Day Seconds Counter (TODSEC)

Base + $2	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read											TODSEC					
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 14-5. TOD Seconds Register (TODSEC)

See Programmer's Sheet on Appendix page B - 97

14.6.4.1 Reserved—Bits 15–6

These bits are reserved or not implemented. They cannot be read nor modified by writing.

14.6.4.2 Time-of-Day Seconds (TODSEC)—Bits 5–0

When TODEN is set, this register is continuously updated to contain the current seconds counter value. The value of the counter can be read through the seconds register, but it cannot be modified. When TODEN is cleared, TOD counting is disabled. This register can then be read or written. When TODEN is set to one again, the counting resumes from the current register value.

14.6.5 Time-of-Day Seconds Alarm Register (TODSAL)

Base + $3	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read											TODSAL					
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 14-6. TOD Seconds Alarm Register (TODSAL)

See Programmer's Sheet on Appendix page B - 98

14.6.5.1 Reserved—Bits 15–6

These bits are reserved or not implemented. They cannot be read nor modified by writing.

14.6.5.2 Time-of-Day Seconds Alarm (TODSAL)—Bits 5–0

When the value contained in this register matches the value of the seconds counter, the seconds alarm is asserted if the Seconds Alarm Enable (TODSA) bit and Alarm Interrupt Enable (TODAEN) bits are set and all other enabled alarm registers also match.

14.6.6 Time-of-Day Minutes Register (TODMIN)

Base + $4	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read											TODMIN					
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 14-7. TOD Minutes Register (TODMIN)

See Programmer's Sheet on Appendix page B - 99

14.6.6.1 Reserved—Bits 15–6

These bits are reserved or not implemented. They cannot be read nor modified by writing.

14.6.6.2 Time-of-Day Minutes (TODMIN)—Bits 5–0

When TODEN is set, this register is continuously updated to contain the current minutes counter value. The value of the counter can be read through the minutes register, but it cannot be modified. When TODEN is cleared, TOD counting is disabled. This register can then be read or written. When TODEN is set to one again, the counting resumes from the current register value.

14.6.7 Time-of-Day Minutes Alarm Register (TODMAL)

Base + $5	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read											TODMAL					
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 14-8. TOD Minutes Alarm Register (TODMAL)

See Programmer's Sheet on Appendix page B - 100

14.6.7.1 Reserved—Bits 15–6

These bits are reserved or not implemented. They cannot be read nor modified by writing.

14.6.7.2 Time-of-Day Minutes Alarm (TODMAL)—Bits 5–0

When the value contained in this register matches the value of the minutes counter, the minutes alarm is asserted if the Minutes Alarm Enable (TODMA) bit and Alarm Interrupt Enable (TODAEN) bit are both set, and all other enabled alarm registers also match the current time.

14.6.8 Time-of-Day Hours Register (TODHR)

Base + $6	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read											TODHR					
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 14-9. TOD Hours Register (TODHR)

See Programmer's Sheet on Appendix page B - 101

14.6.8.1 Reserved—Bits 15–6

These bits are reserved or not implemented. They cannot be read nor modified by writing.

14.6.8.2 Time-of-Day Hours (TODHR)—Bits 5–0

When TODEN is set, this register is continuously updated to contain the current hours counter value. The value of the counter can be read through the hours register, but it can not be modified. When TODEN is cleared, TOD counting is disabled. This register can then be read or written. When TODEN is set to one again, the counting resumes from the current register value.

14.6.9 Time-of-Day Hours Alarm Register (TODHAL)

Base + $7	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read												TODHAL				
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 14-10. TOD Hours Alarm Register (TODHAL)

See Programmer's Sheet on Appendix page B - 102

14.6.9.1 Reserved—Bits 15–5

These bits are reserved or not implemented. They cannot be read nor modified by writing.

14.6.9.2 Time-of-Day Hours Alarm (TODHAL)—Bits 4–0

When the value contained in this register matches the value of the hours counter, the hours alarm is asserted if the Hours Alarm Enable (TODHA) bit and the Alarm Interrupt Enable (TODAEN) bit are both set and all other enabled alarm registers also match the current time.

14.6.10 Time-of-Day Days Register (TODDAY)

Base + $8	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	TODDAY															
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 14-11. TOD Days Register (TODDAY)

See Programmer's Sheet on Appendix page B - 103

14.6.10.1 Time-of-Day Days (TODDAY)—Bits 15–0

When TODEN is set, this counter is continuously updated to contain the current days counter value. The value of the counter can be read through the days register, but it cannot be modified. When TODEN is cleared, TOD counting is disabled. This counter can then be read or written. When TODEN is set to one again, counting will resume counting from the current register value.

14.6.11 Time-of-Day Days Alarm Register (TODDAL)

Base + $9	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	TODDAL															
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 14-12. TOD Days Alarm Register (TODDAL)

See Programmer's Sheet on Appendix page B - 104

14.6.11.1 Time-of-Day Days Alarm (TODDAL)—Bits 15–0

When the value contained in this register matches the value of the days counter, the days alarm interrupt is asserted if the Days Alarm Enable (TODDA) bit and Alarm Interrupt Enable (TODAEN) bit are both set and all other enabled alarm registers also match the current time.

Chapter 15
General Purpose Input/Output (GPIO)

15.1 Introduction

The 5685x General Purpose Input/Output (GPIO) is designed to share package pins with other peripheral modules on the chip. If a peripheral normally controls a given pin is not required, that pin then can be programmed to be a GPIO with programmable pull-up.

15.2 Features

The GPIO module design includes these distinctive features:

- Individual control for each pin to be in either Normal or GPIO mode
- Individual direction control for each pin in GPIO mode
- Individual pull-up enable control for each pin in either Normal or GPIO mode except for MODA, B, and C pins because pull-ups are always disabled

15.3 Block Diagram

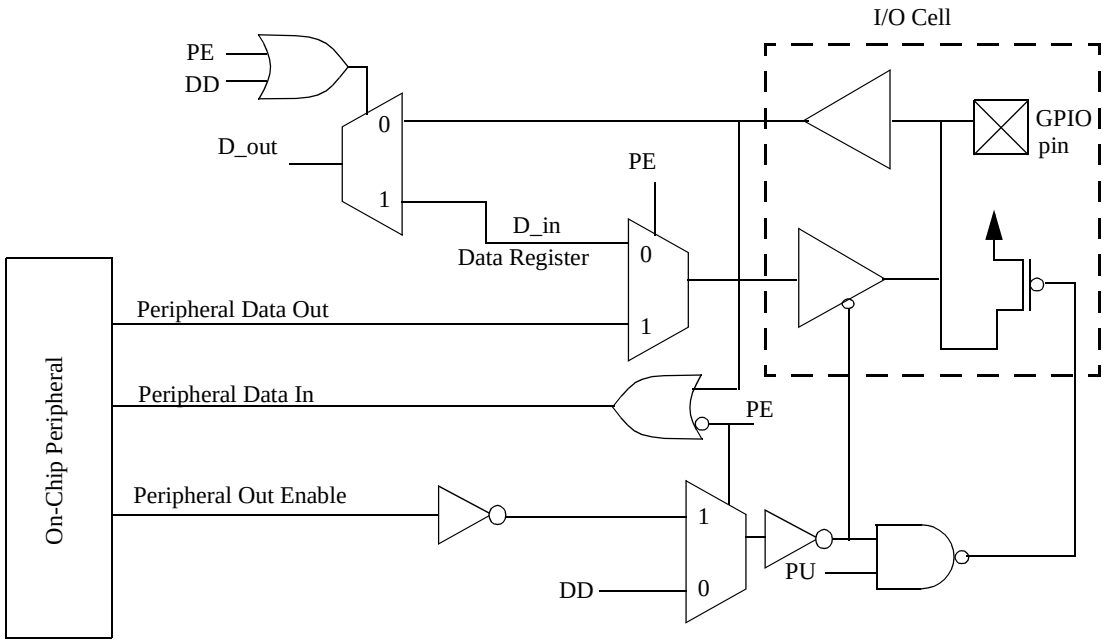


Figure 15-1. Bit-Slice View of GPIO Logic

15.4 Functional Description

Each GPIO pin can be configured as either an input (with or without pull-up) or an output. pull-ups are configured by writing to the Pull-Up Enable (PUE) Registers and are automatically disabled when the pin is being used as an output in either the Normal mode or the GPIO mode.

15.5 Modes of Operation

The GPIO module design contains two major modes of operation.

15.5.1 Normal Mode

This can also be thought of as Peripheral Controlled mode. The peripheral module controls the output enable and any output data to the I/O pad and any input data from the pad is passed to the peripheral. Pull-up enables are controlled by a GPIO register.

15.5.2 GPIO Mode

In this mode, the GPIO module controls the output enable to the pad and supplies any data to be output. Also, any input data can be read from a GPIO memory mapped register. Pull-up enables are controlled by a GPIO register.

In the GPIO mode, the Data Direction Register (DDR) supplies the output enable to the I/O pad to control its direction. The DR supplies the output data if DDR is asserted. The value of the data on the I/O pad can be read by reading Data Register (DR) when DDR is 0. When in GPIO mode the output data from the GPIO to the peripheral module will be driven high and the output data and enable from the peripheral are ignored. The pull-up resistor can be enabled by writing to the PUE Register. The pull-up resistor will be disabled as long as the DDR is set to the Output mode.

15.6 GPIO Configurations

Each GPIO port is controlled by the registers listed in **Table 15-1.** Each register bit corresponds to a GPIO pin. **Section 15-1** illustrates the logic associated with one GPIO bit.

Table 15-1. GPIO Registers Functions

Register	Description	Function
PER	Peripheral Enable Register	Determines if pin functions as GPIO or associated peripheral pin
DDR	Data Direction Register	Determines pin direction (input or output) when pin functions as GPIO
DR	Data Register	Data interface between the GPIO pin and the IPBus
PUER	Pull-up Enable Register	Enables internal pull-up, qualified by other factors

15.7 Module Memory Maps

There are eight GPIO mapped modules listed in the following in tables, **Table 15-2** through **Table 15-9** and their individual accompanying register maps. The GPIO peripherals are summarized in **Figure 15-2** through **Figure 15-9.**

Table 15-2. GPIO A Memory Map (GPIOA_BASE = $1FFE60)

Address Offset	Register Acronym	Register Name	Access Type	Chapter Location
Base + $0	GPIO_A_PER	Peripheral Enable Register	Read/Write	**Section 15.8.1**
Base + $1	GPIO_A_DDR	Data Direction Register	Read/Write	**Section 15.8.9**
Base + $2	GPIO_A_DR	Data Register	Read/Write	**Section 15.8.17**
Base + $3	GPIO_A_PUR	Pull-Up Enable Register	Read/Write	**Section 15.8.25**

Add. Offset	Register Name		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
$0	PER	R	1	1	1	1	1	1	1	1	1	1	1	1	PE			
		W																
$1	DDR	R	0	0	0	0	0	0	0	0	0	0	0	0	DD			
		W																
$2	DR	R	0	0	0	0	0	0	0	0	0	0	0	0	DATA			
		W																
$3	PUR	R	1	1	1	1	1	1	1	1	1	1	1	1	PUE			
		W																

R 0 Read as 0
W Reserved

Figure 15-2. GPIO A Register Map Summary

Table 15-3. GPIO B Memory Map (GPIOB_BASE = $1FFE64)

Address Offset	Register Acronym	Register Name	Access Type	Chapter Location
Base + $0	GPIO_B_PER	Peripheral Enable Register	Read/Write	Section 15.8.2
Base + $1	GPIO_B_DDR	Data Direction Register	Read/Write	Section 15.8.10
Base + $2	GPIO_B_DR	Data Register	Read/Write	Section 15.8.18
Base + $3	GPIO_B_PUR	Pull-Up Enable Register	Read/Write	Section 15.8.26

Add. Offset	Register Name		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
$0	PER	R	PE															
		W																
$1	DDR	R	DD															
		W																
$2	DR	R	DATA															
		W																
$3	PUR	R	PUE															
		W																

R 0 Read as 0
W Reserved

Figure 15-3. GPIO B Register Map Summary

Table 15-4. GPIO C Memory Map (GPIOC_BASE = $1FFE68)

Address Offset	Register Acronym	Register Name	Access Type	Chapter Location
Base + $0	GPIO_C_PER	Peripheral Enable Register	Read/Write	Section 15.8.3
Base + $1	GPIO_C_DDR	Data Direction Register	Read/Write	Section 15.8.11
Base + $2	GPIO_C_DR	Data Register	Read/Write	Section 15.8.19
Base + $3	GPIO_C_PUR	Pull-Up Enable Register	Read/Write	Section 15.8.27

Add. Offset	Register Name		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
$0	PER	R	1	1	1	1	1	1	1	1	1	1	PE					
		W																
$1	DDR	R	0	0	0	0	0	0	0	0	0	0	DD					
		W																
$2	DR	R	0	0	0	0	0	0	0	0	0	0	DATA					
		W																
$3	PUR	R	1	1	1	1	1	1	1	1	1	1	PUE					
		W																

R 0 Read as 0
W Reserved

Figure 15-4. GPIO C Register Map Summary

Table 15-5. GPIO D Memory Map (GPIOD_BASE = $1FFE6C)

Address Offset	Register Acronym	Register Name	Access Type	Chapter Location
Base + $0	GPIO_D_PER	Peripheral Enable Register	Read/Write	Section 15.8.4
Base + $1	GPIO_D_DDR	Data Direction Register	Read/Write	Section 15.8.12
Base + $2	GPIO_D_DR	Data Register	Read/Write	Section 15.8.20
Base + $3	GPIO_D_PUR	Pull-Up Enable Register	Read/Write	Section 15.8.28

Add. Offset	Register Name		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
$0	PER	R	1	1	1	1	1	1	1	1	1	1	PE					
		W																
$1	DDR	R	0	0	0	0	0	0	0	0	0	0	DD					
		W																
$2	DR	R	0	0	0	0	0	0	0	0	0	0	DATA					
		W																
$3	PUR	R	1	1	1	1	1	1	1	1	1	1	PUE					
		W																

R 0 Read as 0
W Reserved

Figure 15-5. GPIO D Register Map Summary

Table 15-6. GPIO E Memory Map (GPIOE_BASE = $1FFE70)

Address Offset	Register Acronym	Register Name	Access Type	Chapter Location
Base + $0	GPIO_E_PER	Peripheral Enable Register	Read/Write	Section 15.8.5
Base + $1	GPIO_E_DDR	Data Direction Register	Read/Write	Section 15.8.13
Base + $2	GPIO_E_DR	Data Register	Read/Write	Section 15.8.21
Base + $3	GPIO_E_PUR	Pull-Up Enable Register	Read/Write	Section 15.8.29

Add. Offset	Register Name		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
$0	PER	R	1	1	1	1	1	1	1	1	1	1	1	1	PE			
		W																
$1	DDR	R	0	0	0	0	0	0	0	0	0	0	0	0	DD			
		W																
$2	DR	R	0	0	0	0	0	0	0	0	0	0	0	0	DATA			
		W																
$3	PUR	R	1	1	1	1	1	1	1	1	1	1	1	1	PUE			
		W																

R 0 Read as 0
W Reserved

Figure 15-6. GPIO E Register Map Summary

Table 15-7. GPIO F Memory Map (GPIOF_BASE = $1FFE74)

Address Offset	Register Acronym	Register Name	Access Type	Chapter Location
Base + $0	GPIO_F_PER	Peripheral Enable Register	Read/Write	Section 15.8.7
Base + $1	GPIO_F_DDR	Data Direction Register	Read/Write	Section 15.8.14
Base + $2	GPIO_F_DR	Data Register	Read/Write	Section 15.8.22
Base + $3	GPIO_F_PUR	Pull-Up Enable Register	Read/Write	Section 15.8.30

Add. Offset	Register Name		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
$0	PER	R	1	1	1	1	1	1	1	1	1	1	1	1	PE			
		W																
$1	DDR	R	0	0	0	0	0	0	0	0	0	0	0	0	DD			
		W																
$2	DR	R	0	0	0	0	0	0	0	0	0	0	0	0	DATA			
		W																
$3	PUR	R	1	1	1	1	1	1	1	1	1	1	1	1	PUE			
		W																

R	0	Read as 0
W		Reserved

Figure 15-7. GPIO F Register Map Summary

Table 15-8. GPIO G Memory Map (GPIOG_BASE = $1FFE78)

Address Offset	Register Acronym	Register Name	Access Type	Chapter Location
Base + $0	GPIO_G_PER	Peripheral Enable Register	Read/Write	Section 15.8.7
Base + $1	GPIO_G_DDR	Data Direction Register	Read/Write	Section 15.8.15
Base + $2	GPIO_G_DR	Data Register	Read/Write	Section 15.8.23
Base + $3	GPIO_G_PUR	Pull-Up Enable Register	Read/Write	Section 15.8.31

Add. Offset	Register Name		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
$0	PER	R	1	1	1	1	1	1	1	1	1	1	1	1	PE			
		W																
$1	DDR	R	0	0	0	0	0	0	0	0	0	0	0	0	DD			
		W																
$2	DR	R	0	0	0	0	0	0	0	0	0	0	0	0	DATA			
		W																
$3	PUR	R	1	1	1	1	1	1	1	1	1	1	1	1	PUE			
		W																

R	0	Read as 0
W		Reserved

Figure 15-8. GPIO G Register Map Summary

Table 15-9. GPIO H Memory Map (GPIOH_BASE = $1FFE7C)

Address Offset	Register Acronym	Register Name	Access Type	Chapter Location
Base + $0	GPIO_H_PER	Peripheral Enable Register	Read/Write	Section 15.8.8
Base + $1	GPIO_H_DDR	Data Direction Register	Read/Write	Section 15.8.16
Base + $2	GPIO_H_DR	Data Register	Read/Write	Section 15.8.24
Base + $3	GPIO_H_PUR	Pull-Up Enable Register	Read/Write	Section 15.8.32

Add. Offset	Register Name		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
$0	PER	R	1	1	1	1	1	1	1	1	1	1	1	1	1	PE		
		W																
$1	DDR	R	0	0	0	0	0	0	0	0	0	0	0	0	0	DD		
		W																
$2	DR	R	0	0	0	0	0	0	0	0	0	0	0	0	0	DATA		
		W																
$3	PUR	R	1	1	1	1	1	1	1	1	1	1	1	1	1	PUE		
		W																

R	0	Read as 0
W		Reserved

Figure 15-9. GPIO H Register Map Summary

15.8 Register Descriptions

Base Addresses:

- GPIOA_BASE = GPIOA_BASE = $1FFE60
- GPIOB_BASE = GPIOB_BASE = $1FFE64
- GPIOC_BASE = GPIOC_BASE = $1FFE68
- GPIOD_BASE = GPIOD_BASE = $1FFE6C
- GPIOE_BASE = GPIOE_BASE = $1FFE70
- GPIOF_BASE = GPIOF_BASE = $1FFE74
- GPIOG_BASE = GPIOG_BASE = $1FFE78
- GPIOH_BASE = GPIOH_BASE = $1FFE7C

15.8.1 Port A Peripheral Enable Register (GPIOA_PER)

Base + $0	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	1	1	1	1	1	1	1	1	1	1	1	1	PE			
Write																
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Figure 15-10. Port A Peripheral Enable Register (GPIOA_PER)

See Programmer's Sheet on Appendix page B - 105

15.8.1.1 Reserved—Bits 15–4

These bits are reserved or not implemented. They are read as 1 and cannot be modified by writing.

15.8.1.2 Peripheral Enable (PE)—Bits 3–0

These bits control whether a given pin is in either Normal or GPIO mode.

- 0 = GPIO mode; pin operation is controlled by GPIO registers
- 1 = Normal mode; pin operation is controlled by the EMI module

15.8.2 Port B Peripheral Enable Register (GPIOB_PER)

Base + $0	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	PE															
Write																
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Figure 15-11. Port B Peripheral Enable Register (GPIOB_PER)

See Programmer's Sheet on Appendix page B - 106

15.8.2.1 Peripheral Enable (PE)—Bits 15–0

These bits control whether a given pin is in either Normal or GPIO mode.

- 0 = GPIO mode; pin operation is controlled by GPIO registers
- 1 = Normal mode; pin operation is controlled by the Host Interface Eight module

15.8.3 Port C Peripheral Enable Register (GPIOC_PER)

Base + $0	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	1	1	1	1	1	1	1	1	1	1	PE					
Write																
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Figure 15-12. Port C Peripheral Enable Register (GPIOC_PER)

See Programmer's Sheet on Appendix page B - 107

15.8.3.1 Reserved—Bits 15–6

These bits are reserved or not implemented. They are read as 1 and cannot be modified by writing.

15.8.3.2 Peripheral Enable (PE)—Bits 5–0

These bits control whether a given pin is in either Normal or GPIO mode.

- 0 = GPIO mode; pin operation is controlled by GPIO registers
- 1 = Normal mode; pin operation is controlled by the ESSI0 module

15.8.4 Port D Peripheral Enable Register (GPIOD_PER)

Base + $0	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	1	1	1	1	1	1	1	1	1	1	PE					
Write																
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Figure 15-13. Port D Peripheral Enable Register (GPIOD_PER)

See Programmer's Sheet on Appendix page B - 108

15.8.4.1 Reserved—Bits 15–6

These bits are reserved or not implemented. They are read as 1and cannot be modified by writing.

15.8.4.2 Peripheral Enable (PE)—Bits 5–0

These bits control whether a given pin is in either Normal or GPIO mode.

- 0 = GPIO mode; pin operation is controlled by GPIO registers
- 1 = Normal mode; pin operation is controlled by the ESSI1 module

15.8.5 Port E Peripheral Enable Register (GPIOE_PER)

Base + $0	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	1	1	1	1	1	1	1	1	1	1	1	1	PE			
Write																
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Figure 15-14. Port E Peripheral Enable Register (GPIOE_PER)

See Programmer's Sheet on Appendix page B - 109

15.8.5.1 Reserved—Bits 15–4

These bits are reserved or not implemented. They are read as 1 and cannot be modified by writing.

15.8.5.2 Peripheral Enable (PE)—Bits 3–0

These bits control whether a given pin is in either Normal or GPIO mode.

- 0 = GPIO mode; pin operation is controlled by GPIO registers
- 1 = Normal mode; pin operation is controlled by the SCI module

15.8.6 Port F Peripheral Enables Register (GPIOF_PER)

Base + $0	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	1	1	1	1	1	1	1	1	1	1	1	1	PE			
Write																
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Figure 15-15. Port F Peripheral Enables Register (GPIOF_PER)

See Programmer's Sheet on Appendix page B - 110

15.8.6.1 Reserved—Bits 15–4

These bits are reserved or not implemented. They are read as 1 and cannot be modified by writing.

15.8.6.2 Peripheral Enable (PE)—Bits 3–0

These bits control whether a given pin is in either Normal or GPIO mode.

- 0 = GPIO mode; pin operation is controlled by GPIO registers
- 1 = Normal mode; pin operation is controlled by the SPI module

15.8.7 Port G Peripheral Enables Register (GPIOG_PER)

Base + $0	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	1	1	1	1	1	1	1	1	1	1	1	1	PE			
Write																
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Figure 15-16. Port G Peripheral Enables Register (GPIOG_PER)

See Programmer's Sheet on Appendix page B - 111

15.8.7.1 Reserved—Bits 15–4

These bits are reserved or not implemented. They are read as 1 and cannot be modified by writing.

15.8.7.2 Peripheral Enable (PE)—Bits 3–0

These bits control whether a given pin is in either Normal or GPIO mode.

- 0 = GPIO mode; pin operation is controlled by GPIO registers
- 1 = Normal mode; pin operation is controlled by the TMR module

15.8.8 Port H Peripheral Enables Register (GPIOH_PER)

Base + $0	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	1	1	1	1	1	1	1	1	1	1	1	1	1	PE		
Write																
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Figure 15-17. Port H Peripheral Enables Register (GPIOH_PER)

See Programmer's Sheet on Appendix page B - 112

15.8.8.1 Reserved—Bits 15–3

These bits are reserved or not implemented. They are read as 1and cannot be modified by writing.

15.8.8.2 Peripheral Enable (PE)—Bits 2–0

These bits control whether a given pin is in either Normal or GPIO mode.

- 0 = GPIO mode; pin operation is controlled by GPIO registers
- 1 = Normal mode; pin operation is controlled by the SIM module

15.8.9 Port A Data Direction Register (GPIOA_DDR)

Base + $1	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	0	0	0	0	0	0	0	0	0	0	0	0	DD			
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 15-18. Port A Data Direction Register (GPIOA_DDR)

See Programmer's Sheet on Appendix page B - 113

15.8.9.1 Reserved—Bits 15–4

These bits are reserved or not implemented. They are read as 0 and cannot be modified by writing.

15.8.9.2 Data Direction (DD)—Bits 3–0

These bits control the pins direction when in GPIO mode. These bits have no effect on the output enables or pull-up enables in the Normal mode.

- 0 = Pin is an input; pull-ups are dependent on value of PUE registers (default)
- 1 = Pin is an output; pull-ups are disabled

15.8.10 Port B Data Direction Register (GPIOB_DDR)

Base + $1	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	DD															
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 15-19. Port B Direction Register (GPIOB_DDR)

See Programmer's Sheet on Appendix page B - 114

15.8.10.1 Data Direction (DD)—Bits 15–0

These bits control the pins direction when in GPIO mode. These bits have no effect on the output enables or pull-up enables in the Normal mode.

- 0 = Pin is an input; pull-ups are dependent on value of PUE registers (default)
- 1 = Pin is an output; pull-ups are disabled

15.8.11 Port C Data Direction Register (GPIOC_DDR)

Base + $1	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	0	0	0	0	0	0	0	0	0	0	DD					
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 15-20. Port C Direction Register (GPIOC_DDR)

See Programmer's Sheet on Appendix page B - 115

15.8.11.1 Reserved—Bits 15–6

These bits are reserved or not implemented. They are read as 0 and cannot be modified by writing.

15.8.11.2 Data Direction (DD)—Bits 5–0

These bits control the pins direction when in the GPIO mode. These bits have no effect on the output enables or pull-up enables in the Normal mode.

- 0 = Pin is an input; pull-ups are dependent on value of PUE registers (default)
- 1 = Pin is an output; pull-ups are disabled

15.8.12 Port D Data Direction Register (GPIOD_DDR)

Base + $1	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	0	0	0	0	0	0	0	0	0	0	DD					
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 15-21. Port D Direction Register (GPIOD_DDR)

See Programmer's Sheet on Appendix page B - 116

15.8.12.1 Reserved—Bits 15–6

These bits are reserved or not implemented. They are read as 0 and cannot be modified by writing.

15.8.12.2 Data Direction (DD)—Bits 5–0

These bits control the pins direction when in the GPIO mode. These bits have no effect on the output enables or pull-up enables in the Normal mode.

- 0 = Pin is an input; pull-ups are dependent on value of PUE registers (default)
- 1 = Pin is an output; pull-ups are disabled

15.8.13 Port E Data Direction Register (GPIOE_DDR)

Base + $1	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	0	0	0	0	0	0	0	0	0	0	0	0	DD			
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 15-22. Port E Direction Register (GIPOE_DDR)

See Programmer's Sheet on Appendix page B - 117

15.8.13.1 Reserved—Bits 15–4

These bits are reserved or not implemented. They are read as 0 and cannot be modified by writing.

15.8.13.2 Data Direction (DD)—Bits 3–0

These bits control the pins direction when in the GPIO mode. These bits have no effect on the output enables or pull-up enables in the Normal mode.

- 0 = Pin is an input; pull-ups are dependent on value of PUE registers (default)
- 1 = Pin is an output; pull-ups are disabled

15.8.14 Port F Data Direction Register (GPIOF_DDR)

Base + $1	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	0	0	0	0	0	0	0	0	0	0	0	0	DD			
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 15-23. Port F Direction Register (GPIOF_DDR)

See Programmer's Sheet on Appendix page B - 118

15.8.14.1 Reserved—Bits 15–4

These bits are reserved or not implemented. They are read as 0 and cannot be modified by writing.

15.8.14.2 Data Direction (DD)—Bits 3–0

These bits control the pins direction when in the GPIO mode. These bits have no effect on the output enables or pull-up enables in the Normal mode.

- 0 = Pin is an input; pull-ups are dependent on value of PUE registers (default)
- 1 = Pin is an output; pull-ups are disabled

15.8.15 Port G Data Direction Register (GPIOG_DDR)

Base + $1	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	0	0	0	0	0	0	0	0	0	0	0	0	DD			
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 15-24. Port G Direction Register (GPIOG_DDR)

See Programmer's Sheet on Appendix page B - 119

15.8.15.1 Reserved—Bits 15–4

These bits are reserved or not implemented. They are read as 0 and cannot be modified by writing.

15.8.15.2 Data Direction (DD)—Bits 3–0

These bits control the pins direction when in the GPIO mode. These bits have no effect on the output enables or pull-up enables in the Normal mode.

- 0 = Pin is an input; pull-ups are dependent on value of PUE registers (default)
- 1 = Pin is an output; pull-ups are disabled

15.8.16 Port H Data Direction Register (GPIOH_DDR)

Base + $1	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	0	0	0	0	0	0	0	0	0	0	0	0	0	DD		
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 15-25. Port H Direction Register (GPIOH_DDR)

See Programmer's Sheet on Appendix page B - 120

15.8.16.1 Reserved—Bits 15–3

These bits are reserved or not implemented. They are read as 0 and cannot be modified by writing.

15.8.16.2 Data Direction (DD)—Bits 2–0

These bits control the pins direction when in the GPIO mode. These bits have no effect on the output enables or pull-up enables in the Normal mode.

- 0 = Pin is an input; pull-ups are dependent on value of PUE registers (default)
- 1 = Pin is an output; pull-ups are disabled

15.8.17 Port A Data Register (GPIOA_DR)

Base + $2	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	0	0	0	0	0	0	0	0	0	0	0	0	DATA			
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 15-26. Port A Data Register (GPIOA_DR)

See Programmer's Sheet on Appendix page B - 121

15.8.17.1 Reserved—Bits 15–4

These bits are reserved or not implemented. They are read as 0 and cannot be modified by writing.

15.8.17.2 Data (DATA)—Bits 3–0

These bits control the output data when in GPIO mode.

15.8.18 Port B Data Register (GPIOB_DR)

Base + $2	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	DATA															
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 15-27. Port B Data Register (GPIOB_DR)

See Programmer's Sheet on Appendix page B - 122

15.8.18.1 Data (DATA)—Bits 15–0

These bits control the output data when in GPIO mode.

15.8.19 Port C Data Register (GPIOC_DR)

Base + $2	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	0	0	0	0	0	0	0	0	0	0	DATA					
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 15-28. Port C Data Register (GPIOC_DR)

See Programmer's Sheet on Appendix page B - 123

15.8.19.1 Reserved—Bits 15–6

These bits are reserved or not implemented. They are read as 0 and cannot be modified writing.

15.8.19.2 Data (DATA)—Bits 5–0

These bits control the output data when in GPIO mode.

15.8.20 Port D Data Register (GPIOD_DR)

Base + $2	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	0	0	0	0	0	0	0	0	0	0	DATA					
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 15-29. Port D Data Register (GPIOD_DR)

See Programmer's Sheet on Appendix page B - 124

15.8.20.1 Reserved—Bits 15–6

These bits are reserved or not implemented. They are read as 0 and cannot be modified by writing.

15.8.20.2 Data (DATA)—Bits 5–0

These bits control the output data when in GPIO mode.

15.8.21 Port E Data Register (GPIOE_DR)

Base + $2	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	0	0	0	0	0	0	0	0	0	0	0	0	DATA			
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 15-30. Port E Data Register (GPIOE_DR)

See Programmer's Sheet on Appendix page B - 125

15.8.21.1 Reserved—Bits 15–4

These bits are reserved or not implemented. They are read as 0 and cannot be modified by writing.

15.8.21.2 Data (DATA)—Bits 3–0

These bits control the output data when in GPIO mode.

15.8.22 Port F Data Register (GPIOF_DR)

Base + $2	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	0	0	0	0	0	0	0	0	0	0	0	0	DATA			
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 15-31. Port F Data Register (GPIOF_DR)

See Programmer's Sheet on Appendix page B - 126

15.8.22.1 Reserved—Bits 15–4

These bits are reserved or not implemented. They are read as 0 and cannot be modified by writing.

15.8.22.2 Data (DATA)—Bits 3–0

These bits control the output data when in GPIO mode.

15.8.23 Port G Data Register (GPIOG_DR)

Base + $2	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	0	0	0	0	0	0	0	0	0	0	0	0	DATA			
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 15-32. Port G Data Register (GPIOG_DR)

See Programmer's Sheet on Appendix page B - 127

15.8.23.1 Reserved—Bits 15–4

These bits are reserved or not implemented. They are read as 0 and cannot be modified by writing.

15.8.23.2 Data (DATA)—Bits 3–0

These bits control the output data when in GPIO mode.

15.8.24 Port H Data Register (GPIOH_DR)

Base + $2	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	0	0	0	0	0	0	0	0	0	0	0	0	0	DATA		
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 15-33. Port H Data Register (GPIOH_DR)

See Programmer's Sheet on Appendix page B - 128

15.8.24.1 Reserved—Bits 15–3

These bits are reserved or not implemented. They are read as 0 and cannot be modified by writing.

15.8.24.2 Data (DATA)—Bits 2–0

These bits control the output data when in GPIO mode.

15.8.25 Port A Pull-Up Enable Register (GPIOA_PUER)

Base + $3	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	1	1	1	1	1	1	1	1	1	1	1	1	PUE			
Write																
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Figure 15-34. Port A Pull-Up Enable Register (GPIOA_PUER)

See Programmer's Sheet on Appendix page B - 129

15.8.25.1 Reserved—Bits 15–4

These bits are reserved or not implemented. They are read as 0 and cannot be modified by writing.

15.8.25.2 Pull-Up Enable (PUE)—Bits 3–0

These bits control whether pull-ups are enabled for inputs in either Normal or GPIO mode.

Pull ups are automatically disabled for outputs in both modes.

- 0 = Pull ups disabled for inputs
- 1 = Pull ups enabled for inputs (default)

15.8.26 Port B Pull-Up Enable Register (GPIOB_PUER)

Base + $3	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	PUE															
Write																
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Figure 15-35. Port B Pull-Up Enable Register (GPIOB_PUER)

See Programmer's Sheet on Appendix page B - 130

15.8.26.1 Pull-Up Enable (PUE)—Bits 15–0

These bits control whether pull-ups are enabled for inputs in either Normal or GPIO mode.

Pull ups are automatically disabled for outputs in both modes.

- 0 = Pull ups disabled for inputs
- 1 = Pull ups enabled for inputs (default)

15.8.27 Port C Pull-Up Enable Register (GPIOC_PUER)

Base + $3	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	1	1	1	1	1	1	1	1	1	1	PUE					
Write																
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Figure 15-36. Port C Pull-Up Enable Register (GPIOC_PUER)

See Programmer's Sheet on Appendix page B - 131

15.8.27.1 Reserved—Bits 15–6

These bits are reserved or not implemented. They are read as 1 and cannot be modified by writing.

15.8.27.2 Pull-Up Enable (PUE)—Bits 5–0

These bits control whether pull-ups are enabled for inputs in either Normal or GPIO mode.

Pull ups are automatically disabled for outputs in both modes.

- 0 = Pull ups disabled for inputs
- 1 = Pull ups enabled for inputs (default)

15.8.28 Port D Pull-Up Enable Register (GPIOD_PUER)

Base + $3	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	1	1	1	1	1	1	1	1	1	1	PUE					
Write																
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Figure 15-37. Port D Pull-Up Enable Register (GPIOD_PUER)

See Programmer's Sheet on Appendix page B - 132

15.8.28.1 Reserved—Bits 15–6

These bits are reserved or not implemented. They are read as 1 and cannot be modified by writing.

15.8.28.2 Pull-Up Enable (PUE)—Bits 5–0

These bits control whether pull-ups are enabled for inputs in either Normal or GPIO mode. Pull ups are automatically disabled for outputs in both modes.

- 0 = Pull ups disabled for inputs
- 1 = Pull ups enabled for inputs (default)

15.8.29 Port E Pull-Up Enable Register (GPIOE_PUER)

Base + $3	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	1	1	1	1	1	1	1	1	1	1	1	1	PUE			
Write																
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Figure 15-38. Port E Pull-Up Enable Register (GPIOE_PUER)

See Programmer's Sheet on Appendix page B - 133

15.8.29.1 Reserved—Bits 15–4

These bits are reserved or not implemented. They are read as 1 and cannot be modified by writing.

15.8.29.2 Pull-Up Enable (PUE)—Bits 3–0

These bits control whether pull-ups are enabled for inputs in either Normal or GPIO mode.

Pull ups are automatically disabled for outputs in both modes.

- 0 = Pull ups disabled for inputs
- 1 = Pull ups enabled for inputs (default)

15.8.30 Port F Pull-Up Enable Register (GPIOF_PUER)

Base + $3	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	1	1	1	1	1	1	1	1	1	1	1	1	PUE			
Write																
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Figure 15-39. Port F Pull-Up Enable Register (GPIOF_PUER)

See Programmer's Sheet on Appendix page B - 134

15.8.30.1 Reserved—Bits 15–4

These bits are reserved or not implemented. They are read as 1 and cannot be modified by writing.

15.8.30.2 Pull-Up Enable (PUE)—Bits 3–0

These bits control whether pull-ups are enabled for inputs in either Normal or GPIO mode.

Pull ups are automatically disabled for outputs in both modes.

- 0 = Pull ups disabled for inputs
- 1 = Pull ups enabled for inputs (default)

15.8.31 Port G Pull-Up Enable Register (GPIOG_PUER)

Base + $3	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	1	1	1	1	1	1	1	1	1	1	1	1	PUE			
Write																
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Figure 15-40. Port G Pull-Up Enable Register (GPIOG_PUER)

See Programmer's Sheet on Appendix page B - 135

15.8.31.1 Reserved—Bits 15–4

These bits are reserved or not implemented.They are read as 1 and cannot be modified by writing.

15.8.31.2 Pull-Up Enable (PUE)—Bits 3–0

These bits control whether pull-ups are enabled for inputs in either Normal or GPIO mode.

Pull ups are automatically disabled for outputs in both modes.

- 0 = Pull ups disabled for inputs
- 1 = Pull ups enabled for inputs (default)

15.8.32 Port H Pull-Up Enable Register (GPIOH_PUER)

Base + $3	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	1	1	1	1	1	1	1	1	1	1	1	1	1	PUE		
Write																
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Figure 15-41. Port H Pull-Up Enable Register (GPIOH_PUER)

See Programmer's Sheet on Appendix page B - 136

15.8.32.1 Reserved—Bits 15–3

These bits are reserved or not implemented. They are read as 1 and cannot be modified by writing.

15.8.32.2 Pull-Up Enable (PUE)—Bits 2–0

This register is not used because pull-ups are always disabled for the MODA, B, and C pins.

15.9 Data Register Access

Care must be taken when accessing the Data Registers. **Table 15-10** summarizes the results of various Data Register accesses in different conditions.

Table 15-10. Data Register Access

Output Enable from Peripheral	PER	DDR	Pad State	Access Type	Data Access Result
X	0	0	Input	Write to DR	Data is written into DR by IPBus. No effect on the pad value.
X	0	1	Output	Write to DR	Data is written into the DR by the IPBus. DR value seen at pad.
X	0	0	Input	Read from DR	Pad state is read by the IPBus. No effect on DR value.
X	0	1	Output	Read from DR	DR value is read by the IPBus. DR value seen at pad.
1	1	X	Input	Write to DR	Data is written into the DR by the IPBus. No effect on pad value.
0	1	X	Output	Write to DR	Data is written into the DR by the IPBus. Peripheral output data is seen at pad.
1	1	X	Input	Read from DR	DR value is read by the IPBus. No effect on the pad or DR value
0	1	X	Output	Read from DR	DR value is read by the IPBus. Peripheral output data is seen at the pad.

15.10 Resets

The GPIO module can only be reset by the $\overline{\text{RST}}$ signal. This forces all registers to their reset state and sets the chip pads to be peripheral controlled with pull-ups enabled.

15.11 Interrupts

The GPIO module does not generate interrupts.

Chapter 16
Host Interface Eight (HI8)

16.1 Introduction

The Host Interface Eight (HI8) is a byte-wide, full-duplex, double-buffered, parallel port able to be connected directly to the data bus of a Host Processor. The HI8 supports a variety of buses, and provides connection with a number of industry standard DSCs, microcomputers, and microprocessors.

Because the host bus can operate asynchronously to the core clock, the HI8 registers are divided into two banks:

1. The Host Side bank is accessible to the external Host
2. The DSC Side bank is accessible to the core

The HI8 supports two classes of interfaces:

1. Host Processor/MCU connection interface
2. General Purpose Input/Output (GPIO) port

HI8 port pins not configured for peripheral use can be configured as GPIO pins.

16.2 Features

16.2.1 Dgital Signal Controller (DSC) Side

- Four internal data I/O-mapped locations
- Sixteen-bit data word
- Transfer modes
  - — DSC-to-Host
  - — Host-to-DSC
  - — Host command
- Handshaking protocols
  - — Software polled
  - — Interrupt driven
  - — DMA accesses.
- Instructions
  - — Memory mapped registers allow the standard MOVE instruction to be used.
  - — Bit addressing instructions (i.e., BFCHG, BFCLR, BFSET, BFTSTH, BFTSTL, BRCLR, BRSET) simplify I/O service routines.

16.2.2 Host Side

- Signals (16 pins)
  - HD7–HD0 data bus
  - HA2-HA0 Host Address bus
  - HRW / $\overline{\text{HRD}}$ - Host Read Write select (HRW) or Host Read ($\overline{\text{HRD}}$) strobe
  - $\overline{\text{HDS}}$ / $\overline{\text{HWR}}$ - Host Data Strobe ($\overline{\text{HDS}}$) or Host Write ($\overline{\text{HWR}}$) strobe
  - $\overline{\text{HCS}}$ Host Chip Select
  - $\overline{\text{HREQ}}$ / $\overline{\text{HTRQ}}$ - Host Request ($\overline{\text{HREQ}}$) or Host Transmit Request ($\overline{\text{HTRQ}}$)
  - $\overline{\text{HACK}}$ / $\overline{\text{HRRQ}}$ - Host Acknowledge ($\overline{\text{HACK}}$) or Host Receive Request ($\overline{\text{HRRQ}}$)
- Mapping
  - Consecutive byte locations
  - Memory or I/O-mapped peripheral for microprocessors, microcontrollers, etc.
- Eight-bit data word
- Transfer modes
  - Mixed 8- and 16-bit data transfers
    - DSC-to-Host
    - Host-to-DSC
  - Host Command
- Handshaking protocols
  - Software polled
  - interrupt driven
- Dedicated interrupts
  - Separate interrupt lines for each interrupt source
  - Special host commands force core interrupts under Host Processor control, useful for:
    - Real-time production diagnostics
    - Debugging window for program development
    - Host control protocols

16.3 Signal Descriptions

16.4 HI8 Host Port

This section's table details the HI8 pins and their functions. Additionally, the HI8 block diagram is located as **Figure 16-1**following the HI8 pins details in **Table 16-1.**

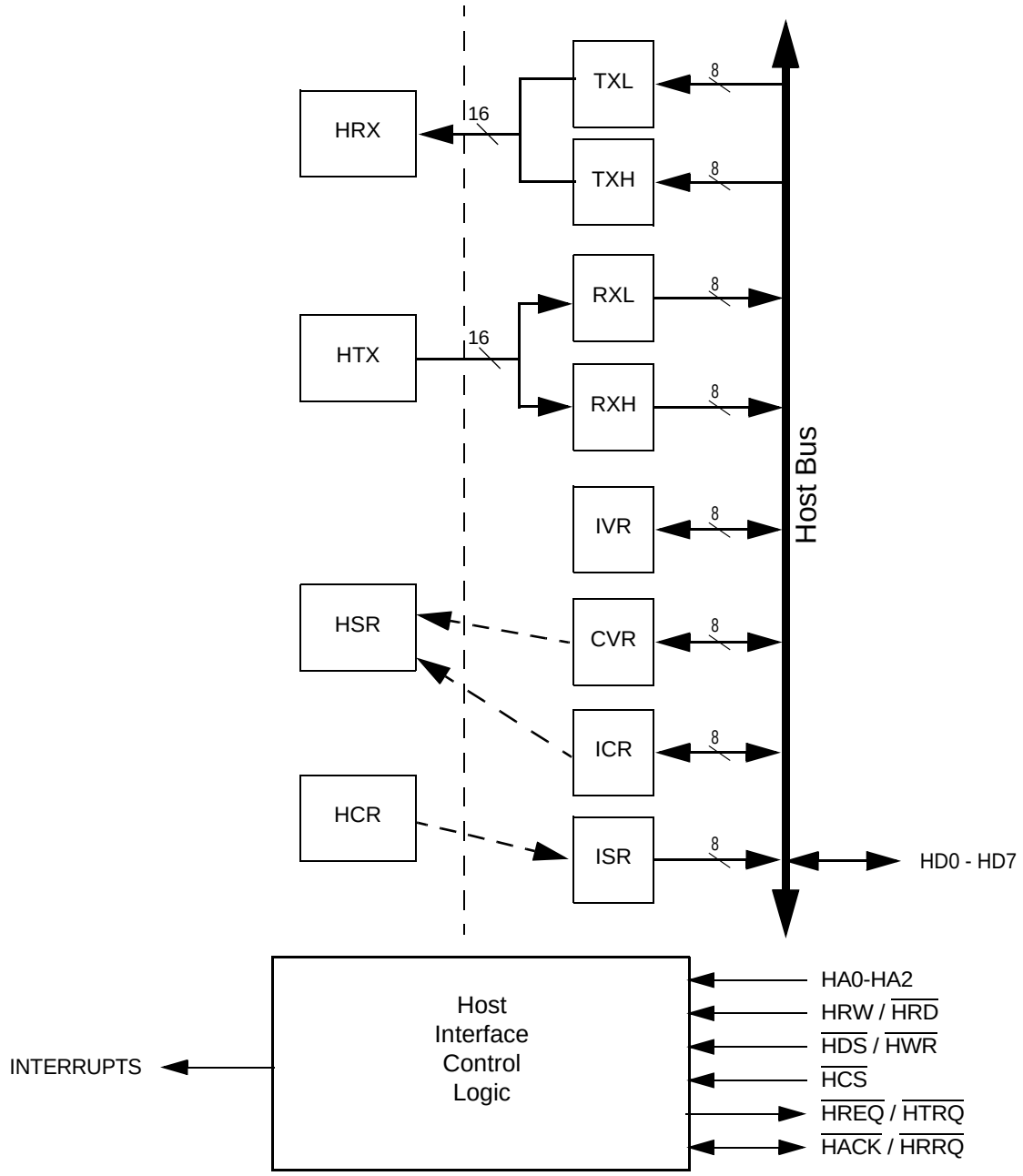
Table 16-1. Host Interface 8 Signals

HI8 Port Pin	Signal Name	Signal Type	State During Reset of Stop	Signal Description
HD[0:7]	HD0 - HD7 or (PB0 - PB7)	Input /Output Input or Output	Disconnected Internally	**Host Data Bus—**This bidirectional data bus is used to transfer data between the Host Processor and the DSC, (HD0-HD7). **Port B 0-7—**These are GPIO pins when not configured for Host port use.
HA[0:2]	HA0 - HA2 or (PB8 - PB10)	Input Input/Output	Disconnected Internally	**Host Address—**These inputs provide the address selection for the HI8 registers, (HA0-HA2). **Port B 8-10—**These are GPIO pins when not configured for Host port use.
$\overline{\text{HRW}}$ / $\overline{\text{HRD}}$	HRW or $\overline{\text{HRD}}$ or (PB11)	Input Input or Output	Disconnected Internally	**Host Read/Write—**When the HI8 is programmed to interface to a single data strobe host bus, this signal is the Read/$\overline{\text{Write}}$ input (HRW). When set low, the Host writes data to DSC. When set high, the Host reads data from DSC. **Host Read Data Strobe—**When the HI8 is programmed to interface to a double data strobe host bus, this signal is the Read Data strobe input ($\overline{\text{HRD}}$), active low. **Port B 11—**This is a GPIO pin when not configured for Host port use.
$\overline{\text{HDS}}$ / $\overline{\text{HWR}}$	$\overline{\text{HDS}}$ or $\overline{\text{HWR}}$ or (PB12)	Input Input Input/Output	Disconnected Internally	**Host Data Strobe—**When the HI8 is programmed to interface to a single-data strobe host bus, this input enables a data transfer on the HI8 when $\overline{\text{HCS}}$ is set to low. **Host Write Data Strobe—**When the HI8 is programmed to interface to a double data strobe host bus and the HI8 function is selected, this active low signal is the Write Data Strobe input ($\overline{\text{HWR}}$). **Port B 12—**This is a GPIO pin when not configured for Host port use.

Table 16-1. Host Interface 8 Signals (Continued)

HI8 Port Pin	Signal Name	Signal Type	State During Reset of Stop	Signal Description
$\overline{\text{HCS}}$	$\overline{\text{HCS}}$ or (PB13)	Input Input/Output	Disconnected Internally	**Host Chip Select—**This is the chip select input for the HI8. **Port B 13—**This is a GPIO pin when not configured for Host port use.
$\overline{\text{HREQ}}$ / $\overline{\text{HTRQ}}$	$\overline{\text{HREQ}}$ or $\overline{\text{HTRQ}}$ or (PB14)	Open Drain Output Open Drain Output Input /Output	Disconnected Internally	**Host Request—**When the HI8 is programmed for HRMS = 0 functionality (typically used on a single data strobe bus), this output is used by the HI8 to request service from the Host Processor. The $\overline{\text{HREQ}}$ may be connected to an interrupt request pin of a Host Processor, a transfer request of a DMA controller, or a control input of external circuitry. **Transmit Host Request—**When the HI8 is programmed for HRMS = 1 functionality (typically used on a double data strobe bus), this signal is the Transmit Host Request output ($\overline{\text{HTRQ}}$). The $\overline{\text{HTRQ}}$ may be connected to an interrupt request pin of a Host Processor. **Port B 14—**This is a GPIO pin when not configured for Host port use.
$\overline{\text{HACK}}$ / $\overline{\text{HRRQ}}$	$\overline{\text{HACK}}$ or $\overline{\text{HRRQ}}$ or (PB15)	Input Open Drain Output Input /Output	Disconnected Internally	**Host Acknowledge—**When the HI8 is programmed for HRMS = 0 functionality (typically used on a single data strobe bus), this input has two functions: 1) provide a host acknowledge signal for DMA transfers, or 2) to control handshaking and provide a host interrupt acknowledge compatible with the *MC68000 family* processors. **Receive Host Request—**When the HI8 is programmed for HRMA = 1 functionality (typically used on a double data strobe bus), this signal is the Receive Host Request output ($\overline{\text{HRRQ}}$). **Port B 15—**This is a GPIO pin when not configured for Host port use.

16.5 HI8 Block Diagram



HCR–Host Control Register
HSR–Host Status Register
HTX–Host Transmit Register
HRX–Host Receive Register
ISR–Interface Status Register
ICR–Interface Control Register

CVR–Command Vector Register
IVR–Interrupt Vector Register
RXH–Receive Register High
RXL–Receive Register Low
TXH–Transmit Register High
TXL–Transmit Register Low

Figure 16-1. HI8 Block Diagram

16.6 DSC Side Register Descriptions (HI8_BASE = $1FFFD8)

HI8 contains four DSC Side Interface registers outlined in **Table 16-2.**

Table 16-2. DSC Side Host Registers (HI_BASE =$1FFFD8)

Address Offset	Register Acronym	Register Name	Access Type	Chapter Location
Base + $0	HCR	Control Register	Read/Write	**Section 16.8.1**
Base + $1	HSR	Status Register	Read/Write	**Section 16.8.2**
Base + $2	HTX	Transmit Data Register	*Write-only*	**Section 16.8.3**
Base + $2	HRX	Receive Data Register	*Read-only*	**Section 16.8.4**

Add. Offset	Register Name		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
$0	HCR	R	0	0	0	0	0	0	0	HRMS	HDDS	TDMAEN	RDMAEN	HF3	HF2	HCIE	HTIE	HRIE
		W	0	0	0	0	0	0	0									
$1	HSR	R	0	0	0	0	0	0	0	0	0	0	HDMA	HF1	HF0	HCP	HTDE	HRDF
		W	0	0	0	0	0	0	0	0	0	0						
$2	HTX	R																
		W	HIGH BYTE (FROM HRX)								LOW BYTE (FROM LRX)							
$2	HRX	R	HIGH BYTE (FROM HTX)								LOW BYTE (FROM LTX)							
		W																

R	0	Read as 0
W		Reserved

Figure 16-2. DSC Host Side Register Map Summary

16.7 Host Side Register Descriptions

HI8 contains eight Host Side Interface registers outlined in **Table 16-3.**

Table 16-3. HI8 Host Side Register Map (HI8 HOST SIDE_BASE = $1FFFD8)

Address Offset	Register Acronym		Register Name	Access Type	Chapter Location
	HLEND = 0 (Big Endian)	HLEND = 1 (Little Endian)			
Base + $0	ICR	ICR	Interface Control	Read/Write	Section 16.9.1
Base + $1	CVR	CVR	Command Vector	Read/Write	Section 16.9.2
Base + $2	ISR	ISR	Interface Status	*Read Only*	Section 16.10.1
Base + $3	IVR	IVR	Interrupt Vector	Read/Write	Section 16.10.2
Base + $4	—	—	Unused	Read as 00	—
Base + $5	—	—	Unused	Read as 00	—
Base + $6	RXH/TXH	RXL/TXL	Receive/Transmit Bytes	Read/Write	Section 16.10.3
Base + $7	RXL/TXL	RXH/TXH	Transmit/Receive Bytes	Read/Write	Section 16.10.4

Add. Offset	Register Name		7	6	5	4	3	2	1	0
$0	ICR	R	INIT	HM1	HM0	HF1	HF0	HLEND	TREQ	RREQ
		W								
$1	CVR	R	HC	HV6	HV5	HV4	HV3	HV2	HV1	HV0
		W								
$2	ISR	R	HREQ	DMA	0	HF3	HF2	TRDY	TXDE	RXDF
		W								
$3	IVR	R	IV7	IV6	IV5	IV4	IV3	IV2	IV1	IV0
		W								

R	0	Read as 0
W		Reserved

Figure 16-3. DSC Host Side Register Map Summary

16.8 DSC Side Registers

The DSC core views the HI8 as a memory mapped peripheral occupying four 16-bit words in data memory space. The DSC can use the HI8 as a normal memory mapped peripheral, using either standard polled or interrupt programming techniques. Separate transmit and receive data registers are double-buffered to allow the DSC and Host Processor to efficiently transfer data at high speed. Memory mapping allows DSC core communication with the HI8 registers to be accomplished using standard instructions and addressing modes. In addition, the MOVE instruction allows HI8-to-memory and memory-to-HI8 data transfers without going through an intermediate register. Both hardware and software reset disable the HI8. These registers can be accessed by the DSC core, or by the external host.

16.8.1 HI8 Control Register (HCR)

The HI8 Control Register (HCR) is 16-bit read/write Control register used by the DSC core to control the HI8 operating mode. Reserved bits are read and written as 0, ensuring future compatibility. **Figure 16-4** illustrates the programming model of the HCR. The initialization values for the HCR bits are described in **Table 16-6.** The HCR bits are described in the following paragraphs.

Base + $0	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	0	0	0	0	0	0	0	HRMS	HDDS	TDMAEN	RDMAEN	HF3	HF2	HCIE	HTIE	HRIE
Write	0	0	0	0	0	0	0									
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 16-4. Host Control Register (HCR)

See Programmer's Sheet on Appendix page B - 137

16.8.1.1 Reserved Bits—Bits 15–9

These bits are reserved or not implemented. They are read as, and written with 0s.

16.8.1.2 Host Request Mode Select (HRMS)—Bit 8

The Host mode Select (HRMS) bit controls the host request pins. **Table 16-4** shows the functionality of the I/O pins controlled by this bit.

Note: The direction of the $\overline{\text{HACK}}$ / $\overline{\text{HRRQ}}$ pin changes based on the HRMS setting. $\overline{\text{HACK}}$ is an input, while $\overline{\text{HRRQ}}$ is an output.

Table 16-4. HRMS Configuration of $\overline{\text{HREQ}}$ and $\overline{\text{HACK}}$ Pins

Pin	HRMS = 0	HRMS = 1
$\overline{\text{HREQ}}$ / $\overline{\text{HTRQ}}$	$\overline{\text{HREQ}}$	$\overline{\text{HTRQ}}$
$\overline{\text{HACK}}$ / $\overline{\text{HRRQ}}$	$\overline{\text{HACK}}$	$\overline{\text{HRRQ}}$

16.8.1.3 Host Dual Data Strobe (HDDS)—Bit 7

When the Host Dual Data Strobe (HDDS) bit is set, the HI8 operates in the Dual Data Strobe Bus mode; a host bus with separated read and write data strobes. When the HDDS bit is cleared, the HI8 operates in the Single Strobe Bus mode, i.e. a Host bus with a single Data Strobe signal. Please see **Figure 16-5** for a description of the two types of buses. The HDDS bit is cleared on hardware reset.

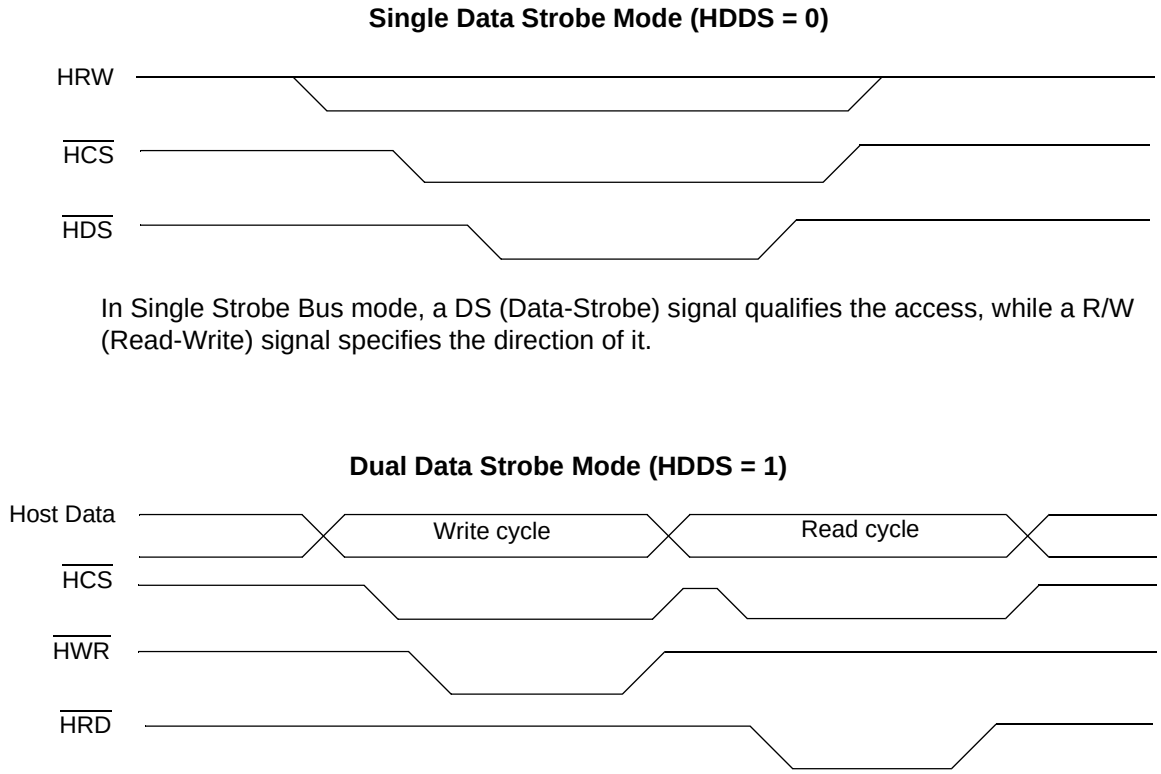


Figure 16-5. Single and Dual Data Strobe Bus Modes

16.8.1.4 DSC Side Transmit DMA Enable (TDMAEN)—Bit 6

The TDMAEN bit is used to enable DSC Side Transmit DMA operations. When this bit is set, the on-chip DMA controller handles transferring data between the HTX register and DSC memory. The on-chip DMA controller must be appropriately configured to implement the desired data transfer.

16.8.1.5 DSC Side Receive DMA Enable (RDMAEN)—Bit 5

The RDMAEN bit is used to enable DSC Side Receive DMA operations. When this bit is set, the on-chip DMA controller handles transferring data between the HRX register and DSC memory. The on-chip DMA controller must be appropriately configured to implement the desired data transfer.

16.8.1.6 Host Flags 2 and 3 (HF2–HF3)—Bits 4–3

The Host Flag 2 and Host Flag 3 (HF2 and HF3) bits are used as general purpose flags for DSC-to-Host communication. HF2 and HF3 may be set or cleared by the core. HF2 and HF3 are reflected in the Interrupt Status Register (ISR) on the Host Side if they are modified by the DSC software, the Host Processor can read the modified values by reading the ISR.

These two flags are not designated for any specific purpose but are general purpose flags. They can be used individually or as encoded pairs in a simple DSC-to-Host communication protocol, implemented in both the DSC and the Host Processor software.

16.8.1.7 Host Command Interrupt Enable (HCIE)—Bit 2

The Host Command Interrupt Enable (HCIE) bit is used to enable a DSC core interrupt when the HCP status bit in the HSR is set. When the HCIE bit is cleared, HCP interrupts are disabled. When the HCIE bit is set, a Host Command Interrupt request occurs if HCP is set. The interrupt address is determined by the Host Command Vector Register (CVR). The HCIE is cleared on hardware reset.

Note: Host interrupt request priorities: If more than one interrupt request source is asserted and enabled (i.e., HRDF = 1, HCP = 1, HRIE = 1, and HCIE = 1) the HI8 generates interrupt requests according to **Table 16-5.**

Table 16-5. HI8 Interrupt Request Order

Priority	Interrupt Source
Highest	Host Command (HCP = 1)
—	Transmit Data (HTDE = 1)
Lowest	Receive Data (HRDF = 1)

16.8.1.8 Host Transmit Interrupt Enable (HTIE)—Bit 1

The Host Transmit Interrupt Enable (HTIE) bit is used to enable a DSC core interrupt when the Host Transmit Data Empty (HTDE) status bit in the HSR is set. When the HTIE bit is cleared, HTDE interrupts are disabled. When the HTIE bit is set, a Host Transmit Data Interrupt request occurs when the HTDE bit is set. The HTIE bit is cleared on hardware reset.

16.8.1.9 Host Receive Interrupt Enable (HRIE)—Bit 0

The Host Receive Interrupt Enable (HRIE) bit is used to enable a DSC core interrupt when the Host Receive Data Full (HRDF) status bit in the Host Status Register (HSR) is set. When the HRIE bit is cleared, HRDF interrupts are disabled. When the HRIE bit is set, a Host Receive Data Interrupt request occurs if the HRDF bit is also set. The HRIE bit is cleared on hardware reset.

16.8.2 HI8 Status Register (HSR)

The 16-bit *read-only* HI8 Status Register (HSR) is used by the DSC to read the status and flags of the HI8 interface. It cannot be directly accessed by the Host Processor. Reserved bits are read as 0s. The value of the HSR after reset is $0002. All bits are cleared, except for the Host Transmit Data Empty (HTDE) bit. It is set. The HSR bits are described in the following paragraphs.

Base + $1	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	0	0	0	0	0	0	0	0	0	0	HDMA	HF1	HF0	HCP	HTDE	HRDF
Write	0	0	0	0	0	0	0	0	0	0						
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

Figure 16-6. Host Status Register (HSR)

See Programmer's Sheet on Appendix page B - 138

16.8.2.1 Reserved—Bits 15–6

These bits are reserved or not implemented. They are read as, and written with 0s.

16.8.2.2 Host DMA Status (HDMA)—Bit 5

The Host DMA Status (HDMA) bit indicates the Host Processor has enabled the Host DMA mode of the HI8 by setting HM1 or HM0 to 1. When the HDMA status bit is set at 0, it indicates the Host DMA mode is disabled by the Host mode bits HM0 and HM1, both having been cleared, in the Interface Control Register ICR and no Host DMA operations are pending. When the HDMA status bit is set, the Host DMA mode is enabled by the Host mode bits HM0 and HM1. The transmit or receive channel not in use can be used by the Host for polled or interrupt operation by the DSC. HDMA is cleared by a DSC reset.

Note: This bit is always 0 if HRMS = 1 since the $\overline{\text{HREQ}}$ and $\overline{\text{HACK}}$ function is disabled and Host DMA operations are disabled.

16.8.2.3 Host Flags 0 and 1 (HF0–HF1)—Bits 4–3

The Host Flag 0-1 (HF0 and HF1) bits are used as a general purpose flags for Host-to-DSC communication. The HF0 and HF1 bits can be set or cleared by the Host. These bits reflect the status of Host Flags HF0 and HF1 in the Interface Control Register (ICR) on the Host Side.

These two flags are not designated for any specific purpose, but are considered general purpose flags. They can be used individually or as encoded pairs in a simple Host-to-DSC communication protocol, implemented in both the DSC and the Host Processor software. The HF0 and HF1 bits are cleared on hardware reset.

16.8.2.4 Host Command Pending (HCP)—Bit 2

The Host Command Pending (HCP) flag bit reflects the status of the HC bit in the Command Vector Register (CVR), indicating a Host Command Interrupt is pending. The HCP bit is set when the HC bit is set, and both bits are cleared by the HI8 hardware when the interrupt request is serviced by the DSC core. The Host can also clear the HC bit, thereby clearing the HCP bit as well. The HCP bit is cleared on hardware reset.

16.8.2.5 Host Transmit Data Empty (HTDE)—Bit 1

The Host Transmit Data Empty (HTDE) flag bit indicates the Host Transmit Data (HTX) register is empty and can be written by the DSC core. The HTDE bit is set when the HTX register is transferred to the RXH/RXL registers, and cleared when Host Transfer Date (HTX) is written by the DSC core. When the HTDE bit is set, the HI8 generates a Transmit Data Full DMA request. HTDE can also be set by the Host Processor using the initialize function. The HTDE bit is set on hardware reset.

16.8.2.6 Host Receive Data Full (HRDF)—Bit 0

The Host Receive Data Full (HRDF) flag bit indicates the Host Receive Data (HRX) register contains data from the Host Processor. The HRDF bit is set when data is transferred from the TXH/TXL registers to the Host Receive Data (HRX) register. The HRDF bit is cleared when the HRX register is read by the DSC core. When the HRDF bit is set, the HI8 generates a receive data full DMA request. The HRDF bit can also be cleared by the Host Processor using the initialize function. The HRDF bit is cleared on hardware reset.

16.8.3 HI8 Transmit Data Register (HTX)

The HI8 Transmit Data (HTX) register is used for DSC-to-Host data transfers. The HTX register is viewed as a 16-bit *write-only* register by the DSC core. Writing to the HTX register clears the HTDE bit in the HSR. The DSC can program the HTIE bit causing a Host Transmit Data interrupt when the HTDE bit is set. The HTX register is transferred as 16-bit data to the receive byte registers RXH/RXL when both the HTDE bit on the DSC Side and the RXDF status bits on the Host Side are cleared. This transfer operation sets both RXDF and HTDE bits. Data should not be written to the HTX register until the HTDE bit is set to prevent the previous data from being overwritten.

Base + $2	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read																
Write	HIGH BYTE (FROM HRX)								LOW BYTE (FROM LRX)							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 16-7. HI8 Transmit Data Register (HTX)

See Programmer's Sheet on Appendix page B - 139

16.8.4 HI8 Receive Data Register (HRX)

The HI8 Receive Data (HRX) register is used for Host-to-DSC data transfers. The HRX register is viewed as a 16-bit *read-only* register by the DSC core. The HRX register is loaded with 16-bit data from the Transmit Data registers TXH/TXL on the Host Side when both the Transmit Data registers empty TXDE on the Host Side, and DSC Host Receive Data Full (HRDF) bits are cleared. This transfer operation sets TXDE and HRDF bits. The HRX register contains valid data when the HRDF bit is set. Reading HRX clears HRDF. The DSC may program the HRIE bit to cause a Host Receive Data interrupt when HRDF is set.

Base + $2	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	HIGH BYTE (FROM HTX)								LOW BYTE (FROM LTX)							
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 16-8. HI8 Receive Data Register (HRX)

See Programmer's Sheet on Appendix page B - 139

16.8.5 DSC Side Registers After Reset

Table 16-6 shows the results of the three reset types on the bits in each of the HI8 registers accessible by the DSC core.

Table 16-6. DSC Side Registers After Reset

Register Name	Register Data	Reset Type		
		Hardware Rest[1]	Software Reset[2]	STOP Reset[3]
HSR	HDMA	0	0	—
	HF1-HF0	0	0	—
	HCP	0	0	0
	HTDE	1	1	1
	HRDF	0	0	0
HRX	HRX[15:0]	Empty	Empty	Empty
HTX	HTX[15:0]	Empty	Empty	Empty

1. Caused by RESET signal
2. Caused by executing the RESET instruction
3. Caused by executing the STOP instruction

16.8.6 HI8 DSC Core Interrupts

The HI8 may request interrupt service from either the DSC core or the Host Processor. The DSC core interrupts are internal and do not require the use of an external interrupt pin. Please refer to **Figure 16-9.** When the appropriate interrupt enable bit in the Host Control Register (HCR) is set, an interrupt condition caused by the Host Processor sets the appropriate bit in the Host Status Register (HSR), thereby generating an interrupt request to the DSC core. The DSC core acknowledges interrupts caused by the Host Processor by jumping to the appropriate interrupt service routine. The three possible interrupts are:

- Receive Data Register Full
- Transmit Data Register Empty
- Host Command

The Host Command can access any interrupt vector in the Interrupt Vector Table, although it has a set of vectors reserved for Host Command use. The DSC interrupt service routine must read or write the appropriate HI8 register. For example, clear the HRDF or HTDE bit to clear the interrupt. In the case of Host Command Interrupts, the interrupt acknowledge from the DSC core Program Controller Unit (PCU) clears the pending interrupt condition.

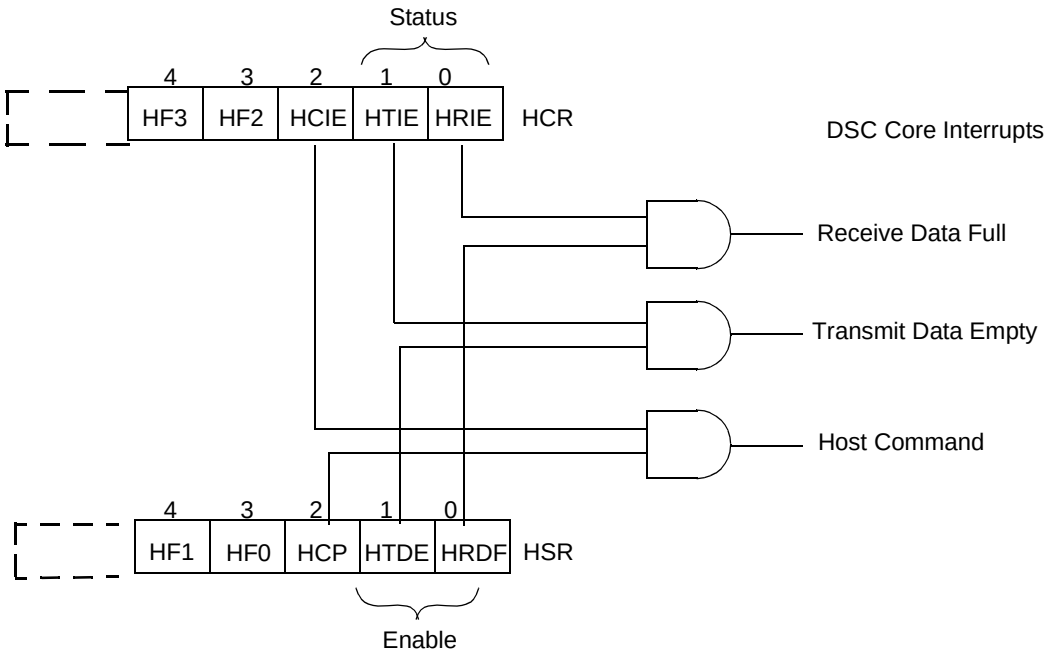


Figure 16-9. HSR–HCR Operation

16.9 Host Side Registers

The HI8 appears to the Host Processor as 8-byte-wide registers. The Host can access the HI8 asynchronously by using polling techniques or interrupt-based techniques. Separate transmit and receive data registers are double buffered to allow the DSC core and Host Processor to transfer data efficiently at High speed.

The HI8 appears to the Host Processor as a memory mapped peripheral occupying eight bytes in the Host Processor address space, detailed in **Figure 16-7.** These registers can be viewed as:

- Interface Control Register (ICR)
- Interface Status Register (ISR)
- Two data registers
  - — Receive Data High/Low (RXH/RXL)
  - — Transmit Data High/Low (TXH/TXL)
- Two vector registers
  - — Interface Vector Receive (IVR)
  - — Command Vector Register (CVR)

The CVR is a special command register used by the Host Processor to issue commands to the DSC. Each of these registers can be accessed only by the Host Processor.

Standard Host Processor instructions such as byte move and addressing modes are used to communicate with the HI8 registers. The HI8 registers are addressed allowing 8-bit Host Processors to use 8/16-bit load and store instructions for data transfers. The $\overline{\text{HREQ}}$ / $\overline{\text{HTRQ}}$ and $\overline{\text{HACK}}$ / $\overline{\text{HRRQ}}$ handshake flags are provided for polled or interrupt driven data transfers with the Host Processor. Because the DSC interrupt response is sufficiently rapid, most host microprocessors can load or store data at their maximum programmed I/O instruction rate without testing the handshake flags for each transfer. If full handshake is not required, the Host Processor can treat the DSC as a fast device, allowing data to be transferred between the Host Processor and the DSC at the fastest host processor data rate.

One of the most innovative features of the HI8 is the Host Command feature. With this feature, the Host Processor can issue vectored interrupt requests to the DSC core. The Host can select any of 128 DSC interrupt routines to be executed by writing a vector address register in the HI8. This flexibility allows the Host programmer to execute as many as 128 pre-programmed functions inside the DSC core. For example, Host Interrupts permits the Host Processor to read or write DSC registers (data or program memory locations), force interrupt handlers (i.e., ESSI, SCI, IRQA and IRQB interrupt routines) to perform control and debugging operations when interrupt routines are implemented in the DSC to perform these tasks.

Note: Please be aware when the DSC core enters the Stop mode, the HI8 pins are electrically disconnected internally, thus disabling the HI8 until the core leaves Stop mode. While the HI8 configuration remains unchanged in Stop mode, the core cannot be restarted via the HI8 interface.

Do not issue a STOP command to the DSC via the HI8 unless some other mechanism for exiting Stop mode is provided.

16.9.1 Interface Control Register (ICR)

The Interface Control Register (ICR) is an 8-bit read/write control register used by the Host Processor to control the HI8 interrupts and flags. The ICR cannot be accessed by the DSC core, but it allows the use of bit manipulation instructions on the Control register. The control bits are described in the following paragraphs. **Figure 16-10** illustrates the programming model of the ICR.

Base + $0	7	6	5	4	3	2	1	0
Read	INIT	HM1	HM0	HF1	HF0	HLEND	TREQ	RREQ
Write								
Reset	0	0	0	0	0	0	0	0

Figure 16-10. Interface Control Register (ICR)

See Programmer's Sheet on Appendix page B - 141

16.9.1.1 Initialize (INIT)—Bit 7

The Initialize (INIT) bit is used by the Host Processor to force initialization of the HI8 hardware. Initialization consists of configuring the HI8 transmit and receive control bits. Using the INIT bit to initialize the HI8 hardware may or may not be necessary, depending upon the software design of the interface.

The type of initialization performed when the INIT bit is set depends on the state of TREQ and RREQ bits located in the Interface Control Register. The INIT command, local to the HI8, is designed to conveniently configure the HI8 into the desired data transfer mode. Those commands are described in **Figure 16-7.** The Host Processor sets the INIT bit, causing the HI8 to execute the INIT command. The interface hardware clears the INIT bit when the command has been executed.

Table 16-7. INIT Execution Definition–Interrupt Mode

TREQ	RREQ	After INIT Execution	Transfer Direction Initialized
0	0	INIT = 0	None
0	1	INIT = 0; RXDF = 0; HTDE = 1	DSC-to-Host
1	0	INIT = 0; TXDE = 1; HRDF = 0	Host-to-DSC
1	1	INIT = 0; RXDF = 0; HTDE = 1;TXDE = 1; HRDF = 0	Host-to/from-DSC

Table 16-8. INIT Execution Definition–HDMA Mode (HM1 = 1)

TREQ	RREQ	After INIT Execution	Transfer Direction Initialized
0	0	INIT = 0; address counter = HM1, HM0	None
0	1	INIT = 0; RXDF = 0; HTDE = 1; address counter = HM1, HM0	DSC-to-Host
1	0	INIT = 0; TXDE = 1; HRDF = 0; address counter = HM1, HM0	Host -to-DSC
1	1	Undefined (illegal)	Host-to/from-DSC

16.9.1.2 Host Mode Control (HM1, HM0)—Bits 6–5

The Host mode control bits HM0 and HM1 select the Transfer mode of the HI8. HM1 and HM0 enable the DMA mode of operation, or they interrupt a no-Host DMA mode of operation when HREQ bit in the Host Control Register (HCR) is set.

When the Host DMA mode is enabled, the $\overline{\text{HREQ}}$ pin is used as a DMA transfer request output to a Host DMA controller and the $\overline{\text{HACK}}$ pin is used as a DMA Transfer Acknowledge input from a

Host DMA controller. The DMA Control bits HM0 and HM1 select the size of the DMA word to be transferred as shown in**Table 16-9.** The direction of the DMA transfer is selected by the TREQ and RREQ bits.

Table 16-9. Mode (HM1, HM0) Bit Definition (HRMS = 0)

HM1	HM0	Mode
0	0	Interrupt mode (HDMA off)
0	1	Illegal
1	0	HDMA mode; 16-bit
1	1	HDMA mode; 8-bit

When both HM1 and HM0 are cleared, the Host DMA mode is disabled and the TREQ and RREQ control bits are used for the Host Processor interrupting via the external Host Request $\overline{\text{HREQ}}$ output pin when the HRMS bit is also cleared. In the Interrupt mode, the Host Acknowledge $\overline{\text{HACK}}$ input pin is used for the *MC68000 family* vectored Interrupt Acknowledge input.

When HM1 is set, the Host DMA mode is enabled and the Host Request ($\overline{\text{HREQ}}$) pin is not available for Host Processor interrupts. When the Host DMA mode is enabled, the TREQ and RREQ bits select the direction of Host DMA transfers; the Host Acknowledge ($\overline{\text{HACK}}$) input pin is used as a Host DMA transfer acknowledge input.

When the Host DMA direction is from DSC-to-Host, the contents of the selected register, RXH or RXL, are enabled onto the Host data bus when the $\overline{\text{HACK}}$ pin is asserted.

If the Host DMA direction is from Host-to-DSC, the contents of the selected register are enabled onto the Host data bus when the $\overline{\text{HACK}}$ pin is asserted.

If the Host DMA direction is from Host-to-DSC, the selected register is written to TXH or TXL from the Host data bus when the $\overline{\text{HACK}}$ pin is asserted.

The size of the Host DMA word to be transferred is determined by the Host mode 0 (HM0) bit. The HI8 register selected during a Host DMA transfer is determined by a 2-bit address counter, preloaded with the value in HM1 and HM0. The address counter substitutes for the Host Address inputs, HA1 and HA0 during a Host DMA transfer. The Host Address input HA2 is forced to 1 during each Host DMA transfer. The address counter can be initialized when the INIT bit is set. After each DMA transfer, the address counter is incriminated to the next register.

When the address counter reaches the highest register (RXL or TXL), the address counter is not incriminated but is loaded with the value in HM1 and HM0. This allows 8- or 16-bit data to be transferred in a circular fashion and eliminates the need for the DMA controller to supply the Host Address HA2, HA1, and HA0 pins. For 16-bit data transfers, the DSC interrupt rate is reduced by a factor of two from the Host Request rate.

HM1 and HM0 are cleared by DSC reset.

Note: When operating in 8-bit HDMA mode, the HLEND affects the location (upper or lower byte) of the 8-bit value in the DSC register (HRX/HTX). For example, in Host-to-DSC transfers, if HLEND = 0 the transferred data is placed in the lower half of the HRX register. If HLEND = 1 the data would appear in the upper half of the register.

Note: Prior to initiating an 8-bit Host-to-DSC transfer the Host should ensure the TXH/TXL registers at address 6 is initialized to 0 because the contents of this register are transferred with the TXL/TXH register to the HRX register.

16.9.1.3 Host Flag 1 (HF1)—Bit 4

The Host Flag 1 (HF1) bit is used as a general purpose flag for Host-to-DSC communication. The HF1 bit can be set or cleared by the Host Processor, but cannot be changed by the DSC core. The HF1 bit is reflected in the HSR on the DSC Side. The HF1 bit is cleared on DSC reset.

16.9.1.4 Host Flag 0 (HF0)—Bit 3

The Host Flag 0 (HF0) bit is used as a general purpose flag for Host-to-DSC communication. The HF0 bit can be set or cleared by the Host Processor, but cannot be changed by the DSC core. The HF0 bit is reflected in the HSR on the DSC Side. The HF0 bit is cleared on DSC reset.

16.9.1.5 ICR Host Little Endian (HLEND)—Bit 2

The Host Little Endian (HLEND) bit allows the HI8 to be accessed by the Host in Little Endian or Big Endian data order. When the HLEND bit is set, the HI8 can be accessed by the Host in *Little Endian* order. The RXH/TXH is located at address $7 and RXL/TXL at $6. When the HLEND bit is cleared, the HI8 can be accessed by the Host in Big Endian Host data order. The RXH/TXH is located at address $6 and RXL/TXL at $7. The HLEND bit is cleared on hardware reset.

16.9.1.6 ICR Transmit Request Enable (TREQ)—Bit 1

The Transmit Request Enable (TREQ) bit is used to control the $\overline{\text{HREQ}}$ pin for Host transmit data transfers. In the Interrupt mode, and the DMA is off, TREQ is used to enable interrupt requests via the external Host Request ($\overline{\text{HREQ}}$ or $\overline{\text{HTRQ}}$) pin when the Transmit Data Register Empty (TXDE) status bit in the Interrupt Status Register (ISR) is set. When TREQ is cleared, TXDE interrupt is disabled. When TREQ is set, the external Host Request $\overline{\text{HREQ}}$ or $\overline{\text{HTRQ}}$ pin is asserted if TXDE is set in the Interrupt mode.

In DMA modes, the TREQ must be set or cleared by software to select the direction of DMA transfers. Setting TREQ sets the direction of the DMA transfer to be from Host-to-DSC and enables the $\overline{\text{HREQ}}$ pin to request these data transfers.

16.9.1.7 Receive Request Enable (RREQ)—Bit 0

This bit is used to control the $\overline{\text{HREQ}}$ pin for Host receive data transfers. In the Interrupt mode (HDMA off), the RREQ is used to enable interrupt requests via the external Host Request ($\overline{\text{HREQ}}$ or $\overline{\text{HRRQ}}$) pin when the Receive Data Register Full (RXDF) status bit in the Interrupt Status register (ISR) is set. When RREQ is cleared, RXDF interrupts are disabled. When RREQ is set, the external Host Request $\overline{\text{HREQ}}$ pin or $\overline{\text{HRRQ}}$ is asserted if RXDF is set in interrupt mode.

In HDMA modes, the RREQ bit must be set or cleared by software to select the direction of DMA transfers. Setting the RREQ bit sets the direction of the HDMA transfer to be from the DSC-to-Host, enabling the $\overline{\text{HREQ}}$ pin to request these data transfers. RREQ is cleared by DSC reset.

Table 16-10,**Table 16-11**, and **Table 16-12** summarizes the effect of RREQ and TREQ on the $\overline{\text{HREQ}}$/$\overline{\text{HTRQ}}$ and $\overline{\text{HACK}}$/$\overline{\text{HRRQ}}$ pins. TREQ is cleared by DSC reset.

Table 16-10. $\overline{\text{HREQ}}$ Pin Definition–Interrupt Mode (HRMS = 0, HM1 = HM0 = 0)

TREQ	RREQ	$\overline{\text{HREQ}}$ Pin
0	0	No interrupts (Polling)
0	1	RXDF Request (Interrupt)
1	0	TXDE Request (Interrupt)
1	1	RXDF and TXDE Request (Interrupt)

Table 16-11. $\overline{\text{HREQ}}$ Pin Definition–Host Mode (HRMS = 0, HM1, HM0 Set for DMA)

TREQ	RREQ	$\overline{\text{RREQ}}$ Pin
0	0	DMA Transfers Disabled
0	1	DSC–> Host Request (RX)
1	0	Host–> DSC Request (TX)
1	1	Undefined (illegal)

Table 16-12. $\overline{\text{HTRQ}}$ and $\overline{\text{HRRQ}}$ Interrupt Mode (HRMS = 1)

TREQ	RREQ	$\overline{\text{HTRQ}}$ Pin	$\overline{\text{HRRQ}}$ Pin
0	0	No Interrupts (Polling)	No Interrupts (Polling)
0	1	No Interrupts (Polling)	RXDF Request (Interrupt)
1	0	TXDE Request (Interrupt)	No Interrupt (Polling)
1	1	TXDE Request (Interrupt)	RXDF Request (Interrupt)

16.9.2 Command Vector Register (CVR)

The Command Vector Register (CVR) is used by the Host Processor, causing the DSC core to execute an interrupt. The Host Command feature is independent of any of the data transfer mechanisms in the HI8. The Host Interface can execute any of the 128 possible interrupt requests to the DSC core. Please refer to the Interrupt Controller (ITCN) chapter for further details.

Base + $1	7	6	5	4	3	2	1	0
Read	HC	HV6	HV5	HV4	HV3	HV2	HV1	HV0
Write								
Reset	0	0	0	0	0	0	0	0

Figure 16-11. Command Vector Register (CVR)

See Programmer's Sheet on Appendix page B - 142

16.9.2.1 Host Command (HC)—Bit 7

This bit is used by the Host Processor to handshake the execution of Host Command interrupts. Normally, the Host Processor sets HC = 1 to request the Host Command interrupt from the DSC core. When the Host Command interrupt is acknowledged by the DSC core the HC bit is cleared by the HI8 hardware. The Host Processor can read the state of the HC bit to determine when the Host Command has been accepted. After writing HC = 1 to the CVR, the Host must not write to the CVR again until the HC bit is cleared by the HI8 hardware. Setting the HC bit causes Host Command pending to be set in the Host Status Register.

16.9.2.2 CVR Host Vector (HV)—Bits 6–0

These seven bits select the Host Command interrupt offset address in the vector table to be used by the Host Command interrupt logic. When the Host Command interrupt is recognized by the DSC interrupt control logic the offset address of the interrupt vector table taken is 2 × HVI0, 6. The Host can write HC and HV in the same write cycle.

The Host Processor can select any of the 128 possible offset addresses of the interrupt vector table in the DSC by writing this address divided by 2, and equal to the vector number, into HV bits. This means the Host Processor can force any of the existing interrupt handlers, SW interrupts such as ESSI, SCI, IRQA, or IRQB and can use any of the reserved or otherwise unused interrupt vectors.

16.10 Servicing the Host Interface

The HI8 can be serviced by using one of the following protocols:

- Polling
- Interrupts
- Host DMA

From the Host Processor viewpoint, the service consists of making a data transfer because this is the only way to reset the appropriate status bits.

16.10.1 Interface Status Register (ISR)

The Interface Status Register (ISR) is an 8-bit *read-only* status register used by the Host Processor to interrogate the status and flags of the HI8. The Host Processor can write this address without affecting the internal state of the HI8, useful to access all of the HI8 registers by stepping through the HI8 addresses. The ISR can not be accessed by the DSC core. The status bits are described in the following paragraphs.

Base + $2	7	6	5	4	3	2	1	0
Read	HREQ	DMA	0	HF3	HF2	TRDY	TXDE	RXDF
Write			0					
Reset	0	0	0	0	0	0	1	0

Figure 16-12. Interface Status Register (ISR)

See Programmer’s Sheet on Appendix page B - 143

16.10.1.1 Host Request (HREQ)—Bit 7

This bit indicates the status of the external Host Request ($\overline{\text{HREQ}}$) output pin if the HRMS bit is cleared; or the external Host Transmit Receive Request ($\overline{\text{HTRQ}}$) output pins, and $\overline{\text{HRRQ}}$ respectively, if HRMS is set. When the HREQ status bit is cleared, it indicates the Host Request pin, $\overline{\text{HREQ}}$ or $\overline{\text{HTRR}}$, $\overline{\text{HTRQ}}$ and $\overline{\text{HRRQ}}$, are deasserted and either Host Processor interrupts or host DMA transfers are being requested.

If the HREQ status bit is set, it means the Host Request ($\overline{\text{HREQ}}$) pin, the Host Transmit Request ($\overline{\text{HTRQ}}$), or Host Receive Request ($\overline{\text{HRRQ}}$) are asserted, indicating the DSC is interrupting the Host Processor or a Host DMA transfer request is being made. The HREQ interrupt request may originate from one or more of two sources:

1. The receive byte registers are full
2. The transmit byte registers are empty

These conditions are indicated by the Interrupt Status Register (ISR) RXDF and TXDE status bits, respectively. If the interrupt source has been enabled by the associated request enable bit in the Interface Control Register (ICR); HREQ, HTRQ, or HRRQ bits is set if one or more of the two enabled interrupt sources is set. DSC reset clears HREQ.

16.10.1.2 Host DMA Status (DMA)–Bit 6

The DMA status bit (DMA) indicates the Host Processor has enabled the DMA mode of the HI8 (HM1 or HM0 =1). When the DMA status bit is clear, it indicates the DMA mode is disabled by the Host mode bits (HM0 and HM1) in the Interface Control Register (ICR) and no DMA operations are pending. When DMA is set, it indicates the DMA mode is enabled and the Host Processor should not use the active DMA channel (RXH:RXL or TXH:TXL depending on DMA direction) to avoid conflicts with the DMA data transfers.

Note: This bit is always 0 if HRMS = 1 in the Host Control Register because the $\overline{\text{HREQ}}$ and $\overline{\text{HACK}}$ function is disabled, thereby disabling the Host DMA operations.

16.10.1.3 Reserved Bit–Bit 5

This bit is reserved or not implemented. It is read as, and written with 0s.

16.10.1.4 Host Flag 3 (HF3)–Bit 4

The Host Flag 3 (HF3) bit in the Interrupt Status Register indicates the state of Host Flag 3 in the Host Control Register on the DSC Side. The HF3 bit can only be changed by the DSC Side. Please see **Figure 16-4.**

16.10.1.5 Host Flag 2 (HF2)—Bit 3

The Host Flag 2 (HF2) bit in the Interface Control Register (ISR) indicates the state of Host Flag 2 in the Host Control Register on the DSCDSC Side. The HF2 bit can only be changed by the DSC Side. Please see **Figure 16-4.**

16.10.1.6 Transmitter Ready (TRDY)—Bit 2

The Transmitter Ready (TRDY) flag bit indicates TXH, TXL, and the HRX registers are empty. When the TRDY bit is set, the data the Host Processor writes to the TXH and TXL registers is immediately transferred to the DSC Side of the HI8. The many applications can use this feature. For example, if the Host Processor issues a Host Command causing the DSC core to read the HRX, the Host Processor can be guaranteed the data transferred to the HI8 is being received by the DSC core.

16.10.1.7 Transmit Data Register Empty (TXDE)—Bit 1

Setting the Transmit Data Register Empty (TXDE) bit indicates the transmit byte registers (TXH, and TXL) are empty and can be written by the Host Processor. TXDE is set when the transmit byte registers are transferred to the HRX register. TXDE is cleared when the transmit TXL/TXH register (based on HLEND setting, the one at address offset seven) is written by the Host Processor. TXDE can be set by the Host Processor using the initialize feature. TXDE may be used to assert the external $\overline{\text{HREQ}}$ pin if the TREQ bit is set. Regardless of whether the TXDE interrupt is enabled, TXDE provides valid status so polling techniques may be used by the Host Processor.

16.10.1.8 Receive Data Register Full (RXDF)—Bit 0

Setting the Receive Data Register Full (RXDF) flag bit indicates the receive byte registers (RXH and RXL) contain data from the DSC Side and can be read by the Host Processor. The RXDF bit is set when the HTX is transferred to the receive byte registers. RXDF is cleared when the receive data RXL/RXH (based on HLEND setting, the one at address offset seven) is read by the Host Processor. RXDF can be used to assert the external $\overline{\text{HREQ}}$ pin if the RREQ bit is set. Regardless of whether the RXDF interrupt is enabled, RXDF provides valid status so polling techniques may be used by the Host Processor.

16.10.2 Interrupt Vector Register (IVR)

The Interrupt Vector Register (IVR) is an 8-bit read/write register typically containing the interrupt vector number used with *MC68000 family* processor vectored interrupts. Only the Host Processor can read and write this register. The contents of IVR are placed on the Host data bus (H0–H7) when both $\overline{\text{HREQ}}$ and $\overline{\text{HACK}}$ pins are asserted and Host DMA is not enabled. The

contents of this register are initialized to a pre-defined value by a hardware or software reset, corresponding to the uninitialized interrupt vector in the *MC68000 family*.

Base + $3	7	6	5	4	3	2	1	0
Read	IV7	IV6	IV5	IV4	IV3	IV2	IV1	IV0
Write								
Reset	0	0	0	0	1	1	1	1

Figure 16-13. Interrupt Vector Register (IVR)

See Programmer's Sheet on Appendix page B - 144

16.10.3 Receive Byte Registers (RXH, RXL)

The Receive Byte Registers are viewed as two 8-bit *read-only* registers by the Host Processor. These registers are called Receive Data High (RXH) and Receive Data Low (RXL). These two registers receive data from the High Byte and Low Byte, respectively from the HTX register. They are selected by three external Host Address (HA2, HA1, and HA0) inputs during a Host Processor read operation or by an on-chip address counter in DMA operations. The Receive Byte Registers contain valid data when the Receive Data Register Full (RXDF) bit is set. The Host Processor may program the RREQ bit to assert the external $\overline{\text{HREQ}}$ pin when the RXDF bit is set. This informs the Host Processor or Host DMA controller of the Receive Byte Registers full condition. Reading the Data Register at Host Address $7 clears the RXDF bit.

- When the HLEND bit in the Interface Control Register (ICR) is *cleared*, the Receive Data Register High (RXH) register is located at address $6 and RXL at $7.
- When the HLEND bit in the ICR is *set*, the RXH register is located at address $7 and the RXL register, at $6.

Base + $6/$7*	7	6	5	4	3	2	1	0
Read	RXH							
Write								
Reset	0	0	0	0	0	0	0	0

Figure 16-14. HI8 Receive Byte High Register (RXH)

See Programmer's Sheet on Appendix page B - 145

* Depends on HLEND setting

Base + $7/$6*	7	6	5	4	3	2	1	0
Read	RXL							
Write								
Reset	0	0	0	0	0	0	0	0

Figure 16-15. HI8 Receive Byte Low Register (RXL)

See Programmer's Sheet on Appendix page B - 145

* Depends on HLEND setting

16.10.4 Transmit Byte Registers (TXH, TXL)

The Transmit Byte Registers are viewed as two 8-bit *write-only* registers by the Host Processor. These registers are called Transmit High (TXH) and Transmit Low (TXL). The two registers transmit data to the High Byte and Low Byte, respectively from the HRX register. They are selected by three external Host Address (HA2, HA1, and HA0) inputs during a Host Processor write operation, or by an on-chip address counter in DMA operations. Data can be written into the Transmit Byte Registers when the Transmit Data Register Empty (TXDE) bit is set. The Host Processor can program the TREQ bit to assert the external $\overline{\text{HREQ}}$ pin when TXDE is set. This informs the Host Processor, or Host DMA controller the Transmit Byte Registers are empty. Writing the data register at Host Address $7 clears the TXDE bit.

- When the HLEND bit in the ICR is *cleared*, the TXH register is located at address $6 and TXL at $7.
- When the HLEND bit in the ICR is *set*, the TXH register is located at address $7 and TXL at $6.

The Transmit Byte Registers are transferred as 16-bit data to the HRX register when both TXDE and the HRDF bits are cleared. This transfer operation sets the TXDE and HRDF bits.

Base + $6/$7*	7	6	5	4	3	2	1	0
Read								
Write	TXH							
Reset	0	0	0	0	0	0	0	0

Figure 16-16. HI8 Transmit Byte High Register (TXH)

See Programmer's Sheet on Appendix page B - 146

* Depends on HLEND setting

Base + $7/$6*	7	6	5	4	3	2	1	0
Read								
Write	TXL							
Reset	0	0	0	0	0	0	0	0

Figure 16-17. HI8 Transmit Byte Low Register (TXL)

See Programmer's Sheet on Appendix page B - 146

* Depends on HLEND setting

16.10.5 Host Side Registers After Reset

Table 16-13 shows the result of the three kinds of reset on bits in each of the HI8 registers seen by the Host Processor. The hardware reset is caused by asserting the RESET pin. Execute the RESET instruction to set the software reset. Execute the STOP instruction to set the Stop reset.

Table 16-13. Host Side Registers After Reset

Register Name	Register Data	Reset Type		
		Hardware Reset	Software Reset	Stop Reset
ICR	All Bits	0	0	—
CVR	HC	0	0	0
	HV[6:0]	$00	$00	—
ISR	HREQ	0	0	1 if TREQ is set, 0 if TREQ is cleared
	DMA	0	0	—
	HF[3:2]	0	0	—
	TRDY	1	1	1
	TXDE	1	1	1
	RXDF	0	0	0
IVR	IV[7:0]	$0F	$0F	—
RX	RXH:RXL	Empty	Empty	Empty
TX	TXH:TXL	Empty	Empty	Empty

16.10.6 HI8 Host Processor Data Transfer

The HI8 looks like static RAM to the Host Processor. To transfer data with the HI8, the Host Processor must:

- Assert the HI8 address to select the register to be read or written
- Select the direction of the data transfer
- Strobe the data transfer

16.10.7 Polling

In the polling mode of operation, the $\overline{\text{HREQ}}$ pin is not connected to the Host Processor and $\overline{\text{HACK}}$ must be negated, ensuring the IVR data is not being driven on H0–H7 when other registers are being polled. $\overline{\text{HACK}}$ can also be configured as a GPIO pin if the $\overline{\text{HACK}}$ function is not required.

Figure 16-18 illustrates the Host Processor first performs a data read transfer to read the ISR to determine, whether:

- RXDF = 1 indicates the Receive Data register is full, and a data read should be performed.
- TXDE = 1 indicates the Transmit Data register is empty, and a data write can be performed.
- When HREQ = 1, the $\overline{\text{HREQ}}$ pin has been asserted, and one of the previous two conditions exist.

Generally, after the appropriate data transfer has been made, the corresponding status bit is updated to reflect the transfer.

If the Host Processor has issued a command to the DSC by writing to the CVR and setting the HC bit, it can read the HC bit in the CVR to determine when the command has been accepted by the interrupt controller in the DSC core. When the command has been accepted for execution, the HC bit is cleared by the interrupt controller in the DSC core.and/or RXDF = 1 and the corresponding enable bit (TREQ or RREQ, respectively) are set. This procedure is depicted in **Figure 16-18.**

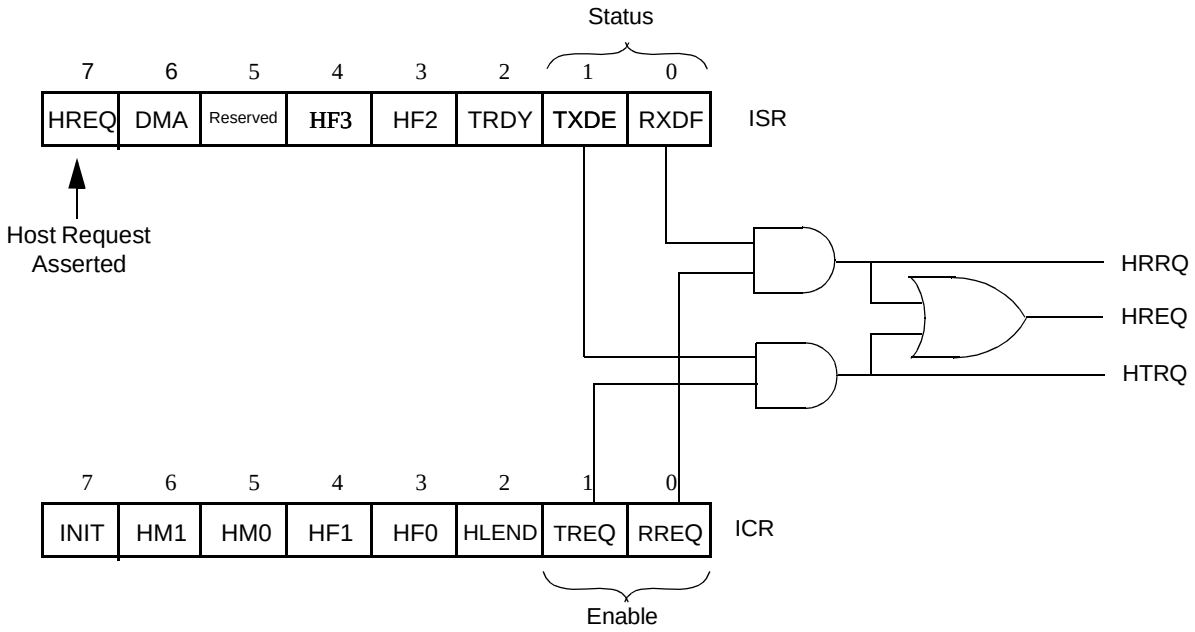


Figure 16-18. HI8 Host Request Structure

16.10.8 Servicing Interrupts

When HREQ is connected to the Host Processor interrupt input, the HI8 can request service from the Host Processor by asserting HREQ. HREQ is asserted when TXDE = 1. Generally, servicing the interrupt begins with reading the Interface Status Register (ISG21R) to determine which DSC flag has generated the interrupt. The Host Processor interrupt service routine must read or write the appropriate HI8 register in order to clear the interrupt. HREQ is deasserted when the enabled request is cleared or masked.

The Host Processor interrupts are external and use the $\overline{\text{HREQ}}$ pin. $\overline{\text{HREQ}}$ is normally connected to the Host Processor maskable interrupt input. The Host Processor acknowledges host interrupts by executing an interrupt service routine. The two LSBs (RXDF and TXDE) of the Interrupt Status Register may be tested by the Host Processor to determine the interrupt source. Please refer to **Figure 16-18.** The Host Processor interrupt service routine must read or write the appropriate HI8 register in order to clear the interrupt. HREQ is deasserted when one of the following occurs:

- The enabled request is cleared or masked
- The DSC is reset

In the case where the Host Processor is a member of the *MC680XX family*, servicing the interrupt starts by asserting HREQ to interrupt the processor. The Host Processor then acknowledges the interrupt by asserting $\overline{\text{HACK}}$. When $\overline{\text{HREQ}}$ and $\overline{\text{HACK}}$ are simultaneously asserted, the contents of the IVR are placed on the Host data bus. This vector tells the Host Processor which routine to use to service the HREQ interrupt.

16.10.9 Host Side DMA Mode Operation

The Host DMA mode allows the transfer of 8-bit or 16-bit data between the DSC HI8 and an external DMA controller. The HI8 provides the synchronization logic between the two asynchronous processor systems. The DSC Side of the interface is serviced by any appropriate servicing mechanism such as polling, interrupts, or DMA transfer.

The external DMA controller provides the transfers between the DSC HI8 registers and the external DMA memory. The external DMA controller must provide the address to the external DMA memory. The address of the selected HI8 register is provided by a DMA address counter in the DSC HI8.

16.10.9.1 Host-to-DSC Host Interface Action

The following four procedure outlines the steps the HI8 hardware takes to transfer DMA data from the Host data bus to DSC memory.

1. Assert the Host Request $\overline{\text{HREQ}}$ output pin when the transmit byte registers TXH/TXL are empty. This always occurs in Host-to-DSC DMA mode when TXDE = 1.
2. Write the selected transmit byte register from the Host data bus when the $\overline{\text{HACK}}$ input pin is asserted by the DMA controller. Deassert the $\overline{\text{HREQ}}$ pin.
3. If the highest register address has not been reached, such as TXDE = 1, post-increment the DMA address counter to select the next register. Wait until $\overline{\text{HACK}}$ is deasserted then go to Step 1.
4. If the highest register address has been reached, such as TXDE = 0, load the DMA address counter with the value in HM1 and HM0 and transfer the transmit byte registers TXH:TXL to the Host Receive Data Register HRX when HRDF = 0. This sets HRDF = 1. Wait until $\overline{\text{HACK}}$ is deasserted then go to Step 1.

Note: The DSC-to-Host data transfers can occur normally in the channel not used for DMA except when the Host must use polling and not interrupts.

Note: The transfer of data from the TXH/TXL register to the HRX register automatically loads the DMA address counter from the HM1 and HM0 bits in the DMA Host-to-DSC mode.

The Host exception is triggered when HRDF = 1. The Host exception routine must read the Host Receive Data Register HRX to clear HRDF.
The transfer from Steps 4 to 1 is automatic if TXDE = 1.

Note: The execution of the Host exception on HRDF = 1 condition occurs after the transfer to Step 1 and is independent of the handshake since it is only dependent on HRDF = 1.

16.10.9.2 Host-to-DSC Host Processor Procedure

The following procedure outlines the typical steps that the Host Processor must take to setup and terminate a Host-to-DSC DMA transfer.

1. Setup the external DMA controller source address, direction, byte count, and other control registers. Enable the DMA controller channel.
2. The DSC must be configured to handle the incoming Host data via polling, interrupts, or DSC Side DMA configuration. If interrupt operation is desired HRIE must be set to enable the HRDF interrupt. If DSC Side DMA operations are desired, set RDMAE to enable DSC transfers when HRDF is set. This could be done with a separate Host Command exception routine in the DSC.
3. Set TXDE and clear HRDF. This can be done with the appropriate Initialize function. The Host must also initialize the DMA counter in the HI8 using the initialize feature. $\overline{\text{HREQ}}$ output pin is asserted immediately by the DSC hardware which begins the DMA transfer.
4. Perform other tasks until interrupted by the DMA controller DMA complete interrupt. The DSC Interface Control Register (ICR), the Interrupt Status Register (ISR), and RXH/RXL may be accessed at any time by the Host Processor (using HA0-HA2, HRW/$\overline{\text{HRD}}$, $\overline{\text{HDS}}$/$\overline{\text{HWR}}$, and $\overline{\text{HCS}}$) but the transmit byte registers (TXH/TXL) may not be accessed until the DMA mode is disabled.
5. Terminate the DMA controller channel to disable DMA transfers.
6. Terminate the DSC HI8 DMA mode by clearing the HM1 and HM0 bits and clearing TREQ in the Interface Control Register (ICR).

16.10.9.3 DSC-to-Host Interface Action

The following procedure outlines the steps that the HI8 hardware takes to transfer DMA data from DSC memory to the Host data bus.

1. The transmit exception is triggered when HTIE = 1 and HTDE = 1, for Interrupt mode transfers, or when TDMAE = 1 and HTDE = 1, for on-chip DMA transfers. The exception routine software, or on-chip DMA, writes the data word into HTX.

2. Transfer the HTX register to the receive byte registers RXH/RXL when they are empty (RXDF = 0). This automatically occurs. Load the Host DMA address counter from HM1 and HM0. This action sets HTDE = 1 and trigger another DSC transmit exception to write HTX.
3. Assert the Host Request ($\overline{\text{HREQ}}$) pin when the receive byte registers are full.
4. Enable the selected Receive Byte register on the Host data bus when $\overline{\text{HACK}}$ is asserted. Deassert the Host Request $\overline{\text{HREQ}}$ pin.
5. If the highest register address has not been reached (i.e., RXDF = 1), post-increment the HI8 DMA address counter to select the next register. Wait until $\overline{\text{HACK}}$ is deasserted, then go to Step 3.
6. If the highest register address has been reached (i.e., RXDF = 0), wait until $\overline{\text{HACK}}$ is deasserted then go to step 2. The DSC transmit exception must have written HTX (i.e., HTDE = 0) before Step 2 is executed.

Note: The HOST –> DSC data transfers can occur normally in the channel not used for DMA except when the Host must use polling and not interrupts.

Note: The transfer of data from the HTX register to the RXH/RXL registers automatically loads the HI8 DMA address counter from the HM1 and HM0 bits when in DMA DSC-to-Host mode.

16.10.9.4 DSC-to-Host Processor Procedure

The following procedure outlines the typical steps the Host Processor must take to setup and terminate a DSC-to-Host DMA transfer.

1. Setup the external DMA controller destination address, direction, byte count, and other control registers. Enable the DMA controller channel.
2. The DSC must be configured to provide the outgoing data via polling, interrupts, or DSC Side DMA configuration. If interrupt operation is desired HTIE must be set to enable the HTDE interrupt. If DSC Side DMA operations are desired, set TDMAE to enable DSC transfers when HTDE is set. This could be done with a separate Host Command exception routine in the DSC.
3. Set HTDE and clear RXDF. This can be done with the appropriate INIT function. The DSC Host transmit exception is activated immediately by DSC hardware which begins the DMA transfer.
4. Perform other tasks until interrupted by the DMA controller DMA complete interrupt. The DSC Interface Control Register (ICR), the Interrupt Status Register (ISR), and TXH/TXL may be accessed at any time by the Host Processor (using HA0-HA2, HRW/$\overline{\text{HRD}}$,

$\overline{\text{HDS}}$/$\overline{\text{HWR}}$, and $\overline{\text{HCS}}$) but the receive byte registers (RXH and RXL) may not be accessed until the DMA mode is disabled.

5. Terminate the DMA controller channel to disable DMA transfers.
6. Terminate the DSC HI8 DMA mode by clearing the HM1 and HM0 bits and clearing RREQ in the Interface Control Register (ICR).

16.10.10 Host Port Use Considerations

Careful synchronization is required when reading multi-bit registers written by another asynchronous system. This is a common problem when two asynchronous systems are connected. The situation exists in the Host port. However, if the port is used in the way it was designed, proper operation is guaranteed. The considerations for proper operation are discussed below.

16.10.10.1 Host Programmer Considerations

1. Unsynchronized Reading of receive byte registers.

   When reading receive byte registers, RXH or RXL, the Host programmer should use interrupts or poll the RXDF flag indicating data is available. This guarantees the data in the receive byte registers are stable.

2. Overwriting transmit byte registers.

   The Host programmer should not write to the transmit byte registers, TXH or TXL, unless the TXDE bit is set, indicating the transmit byte registers are empty. After writing, time must be allowed for TXDE to update. From the rising edge of the clock effecting write of the TX register pair high order byte (address 0x7) to the valid indication of TXDE is a period of two DSC clock cycles. This guarantees the DSC will read stable data when it reads the HRX register.

3. Synchronization of Status Bits from DSC-to-Host.

   HC, HREQ, DMA, HF3, HF2, TRDY, TXDE, and RXDF status bits are set or cleared from inside the DSC and read by the Host Processor. The Host can not read these status bits very quickly without regard to the clock rate used by the DSC there is a chance the state of the bit could be changing during the read operation. This is generally not a system problem because the bit is read correctly in the next pass of any Host polling routine. However, if the Host holds the $\overline{\text{HCS}}$ input pin for the minimum assert time plus 1.5 Clock cycles, the status data is guaranteed to be stable. The 1.5 Clock cycles is first used to synchronize the $\overline{\text{HCS}}$ signal and then to block internal updates of the status bits. There is no other minimum $\overline{\text{HCS}}$ assert time relationship to the DSC clocks. There is a minimum $\overline{\text{HCS}}$ deassert time of 1.5 Clock cycles so that the blocking latch can be deasserted to allow updates if the Host is in a tight polling loop. This only applies to reading status bits.

The only potential system problem with the uncertainty of reading any status bits by the Host is HF3 and HF2 as an encoded pair. For example, if the DSC changes HF3 and HF2 from 00 to 11, there is a very small probability the Host could read the bits during the transition and receive 01 or 10 instead of 11. If the combination of HF3 and HF2 has significance, the Host would potentially read the wrong combination.

Solutions:

- Read the bits twice and check for consensus.
- Assert $\overline{\text{HCS}}$ access for $\overline{\text{HCS}}$ + 1.5 Clock cycles so the status bit transitions are stabilized.

4. Overwriting the Host Vector

   The Host programmer should change the Host Vector register only when the Host Command (HC) bit is clear. This guarantees the DSC interrupt control logic receives a stable vector.

5. Cancelling a pending Host Command Exception

The Host Processor may elect to clear the HC bit to cancel the Host Command exception request at any time before it is recognized by the DSC. The DSC may execute the Host exception after the HC bit is cleared:

- Because the Host does not know exactly when the exception is recognized
- Because of synchronization
- Because of exception processing pipelining. As a result, the HV must not be changed at the same time the HC bit is cleared. In this way, if the exception was taken, the vector is known.

16.10.10.2 DSC Programmer Considerations

Reading HF1 and HF0 as an encoded pair:

- DMA
- HF1
- HF0
- HCP
- HTDE
- HRDF

Chapter 17
JTAG Port

17.1 Introduction

This chapter describes the 56F800E core-based family of chips, providing board and chip-level debugging and high-density circuit board testing specific to Joint Test Action Group (JTAG).

The 5685x provides board and chip-level testing capability through two on-chip modules, both accessed through the JTAG port/EOnCE module interface:

- Enhanced On-chip Emulation (EOnCE) module
- Test Access Port (TAP) and 16-state controller, also known as the JTAG port

Presence of the JTAG port/EOnCE module interface permits insertion of the DSC chip into a target system while retaining debug control. This capability is especially important for devices without an external bus, because it eliminates the need for an expensive cable to bring out the chip footprint required by a traditional emulator system.

The Enhanced OnCE (EOnCE) module is used in Digital Signal Controller (DSC) chips to debug application software employed with the chip. The port is a separate on-chip block allowing non-intrusive DSC interaction with accessibility through the pins of the JTAG interface. The EOnCE module makes it possible to examine registers, memory, or on-chip peripherals' contents in a special debug environment. This avoids sacrificing any user-accessible, on-chip resources to perform debugging procedures. Please refer to the *DSP56F800E Core-Based Reference Manual* (DSP56800ERM)for details about implementation of the 5685x EOnCE module.

The JTAG port is a dedicated user-accessible TAP compatible with the *IEEE 1149.1a-1993 Standard Test Access Port and Boundary Scan Architecture*. Problems associated with testing high-density circuit boards have led to the development of this proposed standard under the sponsorship of the Test Technology Committee of IEEE and the JTAG. 5685x supports circuit board test strategies based on this standard.

Six dedicated pins interface to the TAP containing a 16-state controller. The TAP uses a boundary scan technique to test the interconnections between integrated circuits after they are assembled onto a Printed Circuit Board (PCB). Boundary scans allow observation and control signal levels at each component pin through a Shift register placed next to each pin. This is important for testing continuity and determining if pins are stuck at a one or zero level.

17.2 Features

Features of the Test Access Port (TAP) port include:

- Perform boundary scan operations to test circuit board electrical continuity
- Bypass the TAP for a given circuit board test by replacing the Boundary Scan Register (BSR) with a single-bit register
- Sample system pins during operation and transparently shift-out the results in the BSR
- Preload output pins prior to invoking the EXTEST instruction
- Disable the output drive to pins during circuit board testing
- Provide a means of accessing the EOnCE module controller and circuits to control a target system
- Query the IDCODE from any TAP in the system
- Force test data onto the peripheral outputs while replacing its BSR with a single bit register
- Enable/disable pull-up devices on peripheral boundary scan pins

17.3 Master Test Access Port (TAP)

The Master TAP consists of:

- Synchronous finite 16-bit state machine
- Eight-bit Instruction Register (IR)
- Chip Identification (CID) register
- Bypass (BYPASS) register
- Boundary Scan Register (BSR)

Please see **Figure 17-1** for additional information.

17.3.1 Signal Description

As described in IEEE 1149.1a, the JTAG port requires a minimum of four pins to support TDI, TDO, TCK, and TMS signals. The 5685x also uses the optional $\overline{\text{TRST}}$ input signal and the $\overline{\text{DE}}$ output signal used by the EOnCE module interface. Pin functions are described in **Table 17-1.**

Table 17-1. JTAG Pin Descriptions

Pin Name	Pin Description
TDI	**Test Data Input**—This input pin provides a serial input data stream to the JTAG and the EOnCE modules. It is sampled on the rising edge of TCK and has an on-chip pull-up resistor.
TDO	**Test Data Output**—This tri-state output pin provides a serial output data stream from the JTAG and the EOnCE modules. It is driven in the Shift-IR and Shift-DR controller states of the JTAG state machine and changes on the falling edge of TCK.
TCK	**Test Clock Input**—This input pin provides the clock to synchronize the test logic and shift serial data to and from all TAP Controllers and the TLM. If the EOnCE module is not being accessed using the Master TAP Controllers, the maximum TCK frequency is 1/4 the maximum frequency for the 56800E core. When accessing the EOnCE module through the 56800E core TAP Controller, the maximum frequency for TCK is 1/8 the maximum frequency for the 56800E core. The TCK pin has an on-chip pull-down resistor.
TMS	**Test Mode Select Input**—This input pin is used to sequence the JTAG TAP Controller's state machine. It is sampled on the rising edge of TCK and has an on-chip pull-up resistor.
$\overline{\text{TRST}}$	**Test Reset**—This input provides a reset signal to the JTAG TAP Controller. The $\overline{\text{TRST}}$ pin has an on-chip pull-up resistor.
$\overline{\text{DE}}$	**Debug Event**—This output signal debugs events detected on a trigger condition.

17.4 TAP Block Diagram

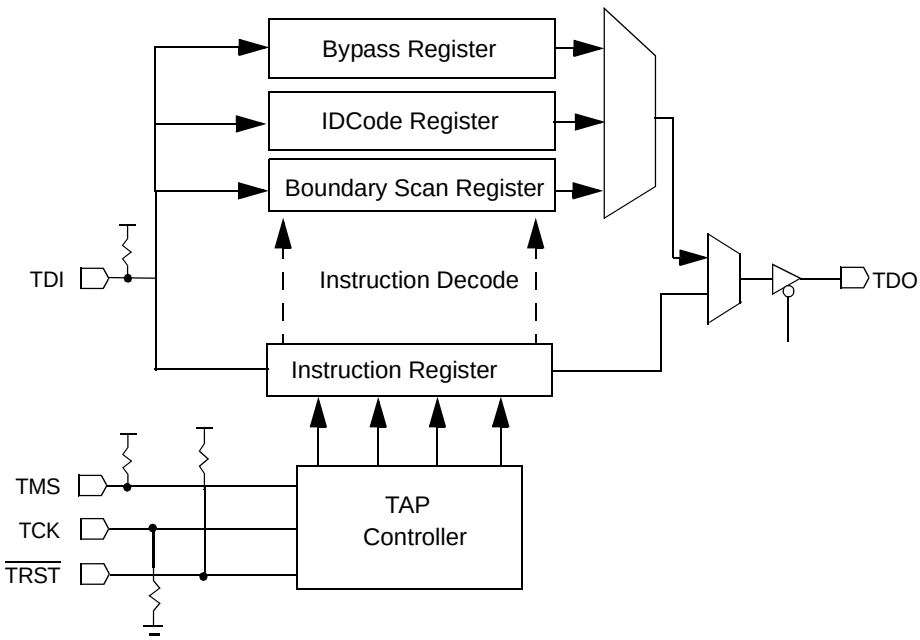


Figure 17-1. Test Access Port (TAP) Block Diagram

17.5 JTAG Port Architecture

The TAP Controller is a simple state machine used to sequence the JTAG port through its varied operations:

- Serially shift in or out a JTAG port command
- Update and decode the JTAG port Instruction Register (JTAGIR)
- Serially input or output a data value
- Update a JTAG port or EOnCE module register

Note: The JTAG port supervises the shifting of data into and out of the EOnCE module through TDI and TDO pins respectively. In this case, the shifting is guided by the same controller used when shifting JTAG information.

The JTAG block diagram is illustrated in **Figure 17-1.** The JTAG port has four read/write registers:

1. Instruction Register (JTAGIR)
2. Chip Identification (CID) register
3. Bypass Register (JTAGBR)
4. Boundary Scan Register (BSR)

Access to the EOnCE registers is described in the *DSP56800E Reference Manual*.

17.5.1 JTAG Instruction Register (JTAGIR) and Decoder

The TAP Controller contains a 8-bit instruction register. The instruction is presented to an instruction decoder during the update-instruction register state. Please see **Section 17.8** for a description of the TAP Controller operating states. The instruction decoder interprets and executes the instructions according to the conditions defined by the TAP Controller state machine.

The 5685x includes the three mandatory public instructions:

1. BYPASS
2. SAMPLE/PRELOAD
3. EXTEST

The 5685x includes four public instructions:

1. CLAMP
2. HIGHZ
3. IDCODE
4. TLM_SELECT

The eight bits B[7:0] of the IR, decode the nine instructions, illustrated in **Figure 17-2** and its data provided in **Table 17-2.** All other encodings are reserved.

IR	7	6	5	4	3	2	1	0‘
Read/Write	B7	B6	B5	B4	B3	B2	B1	B0
Reset	0	0	0	0	0	0	1	0

Figure 17-2. JTAGIR Register

CAUTION

Reserved JTAG instruction encodings should not be used. Hazardous operation of the chip could occur if these instructions are used.

Table 17-2. Master TAP Instructions Opcode

Instruction	Target Register	Opcode
EXTEST	Boundary	00000000
BYPASS	Bypass	11111111
SAMPLE_PRELOAD	Boundary	00000001
IDCODE	IDCode	00000010
TLM_SEL	TLM	00000101
HIGHZ	Bypass	00000110
CLAMP	Bypass	00000111
Reserved	Reserved	00000011
Reserved	Reserved	00000100
Reserved	Reserved	00001000

17.5.1.1 External Test Instruction (EXTEST)

The External Test (EXTEST) instruction enables the BSR between TDI and TDO, including cells for all digital device signals and associated control signals. The EXTAL and $\overline{\text{RESET}}$ pins, and any codec pins associated with analog signals, are not included in the BSR path.

In EXTEST, the BSR is capable of scanning user-defined values onto output pins, capturing values presented to input signals, and controlling the direction and value of bidirectional pins. EXTEST instruction asserts internal system reset for the DSC system logic during its run in order to force a predictable internal state while performing external boundary scan operations.

17.5.1.2 Bypass Instruction (BYPASS)

The BYPASS instruction enables the single-bit bypass register between TDI and TDO, illustrated in **Figure 17-3.** This creates a Shift register path from TDI to the bypass register and finally to TDO, circumventing the BSR. This instruction is used to enhance test efficiency by shortening the overall path between TDI and TDO when no test operation of a component is required. In this instruction, the system logic is independent of the TAP. When this instruction is selected, the test logic has no effect on the operation of the on-chip system logic, required in IEEE 1149.1-1993a.

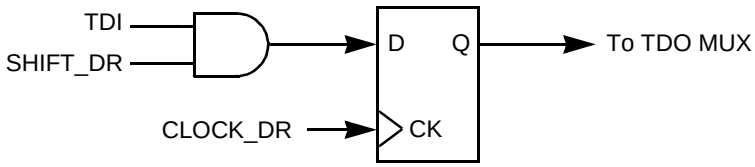


Figure 17-3. Bypass Register Diagram

17.5.2 Sample and Preload Instructions (SAMPLE/PRELOAD)

The SAMPLE/PRELOAD instruction enables the BSR between TDI and TDO. When this instruction is selected, the test logic operation has no effect on the operation of the on-chip system logic. Nor does it have an effect on the flow of a signal between the system pin and the on-chip system logic, specified by IEEE 1149.1-1993a. This instruction provides two separate functions.

1. First, it provides a means to obtain a snapshot of system data and control signals (SAMPLE). The snapshot occurs on the rising edge of TCK in the Capture-DR controller state. The data can be observed by shifting it transparently through the BSR.

   In a normal system configuration, many signals require external pull-ups assuring proper system operation. Consequently, the same is true for the SAMPLE/PRELOAD

functionality. Data latched into the BSR during the Capture-DR controller state may not match the drive state of the package signal if the system requiring pull-ups are not present within the test environment.

2. The second function of the SAMPLE/PRELOAD instruction is to initialize the BSR output cells (PRELOAD) prior to selection of the CLAMP or EXTEST instruction. This initialization ensures known data appears on the outputs when executing EXTEST. The data held in the Shift register stage is transferred to the output latch on the falling edge of TCK in the update Data Register (DR) controller state. Data is not presented to the pins until the CLAMP or EXTEST instruction is executed.

Note: Since there is no internal synchronization between the JTAG clock (TCK) and the system Clock (CLK), some form of external synchronization to achieve meaningful results when sampling system values using the SAMPLE/PRELOAD instruction must be provided.

17.5.2.1 Identification Code Instruction (IDCODE)

The IDCODE instruction enables the IDREGISTER between TDI and TDO. It is provided as a public instruction to allow the manufacturer part number and version of a component to be determined through the TAP.

17.5.2.2 TAP Linking Module Select (TLM_SEL)

TLM_SEL instruction is a user-defined JTAG instruction. It is used to disable the Master TAP and enable the TAP Linking Module (TLM). The TLM provides a means of connecting one or more TAPs in a multi-TAP design, responding to the IC's test pins in IEEE 1149.1 scan operations. TLM serves as a community data register used to set the TAP linking configuration desired. The TLM register is a 4-bit register, illustrated in **Figure 17-3,** and enabled between TDI and TDO during a shift Data Register (DR) operation. It is updated on the Update DR operation.

Table 17-3. TLM Register

Update DR (Load)	Shift DR (Capture)	Bit
Master TAP	N/A	0
56800E TAP	N/A	1
N/C	N/A	2
N/C	N/A	3

17.5.2.3 High Z Instruction (HIGHZ)

The HIGHZ instruction enables the single-bit bypass register between TDI and TDO. It is provided as a public instruction in order to prevent having to drive the output signals back during circuit board testing. When the HIGHZ instruction is invoked, all output drivers are placed in an inactive-drive state. HIGHZ asserts internal system reset for the system logic for the duration of HIGHZ in order to force a predictable internal state while performing external boundary scan operations.

17.5.3 JTAG Chip Identification (CID) Register

The Chip Identification (CID) register is a 32-bit register providing a unique JTAG ID for the DSP56853/54/55/57/58. It is offered as a public instruction to allow the manufacturer, part number, and version of a component to be determined through the TAP. **Figure 17-4** illustrates the CID register configuration.

CIR = $2	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	PNUM 3	PNUM 2	PNUM 1	PNUM 0	MFG ID 11	MFG ID 10	MFG ID 9	MFG ID 8	MFG ID 7	MFG ID 6	MFG ID 5	MFG ID 4	MFG ID 3	MFG ID 2	MFG ID 1	MFG ID 0
Write																
Reset	1	0	1	0	0	0	0	0	0	0	0	1	1	1	0	1

CIR = $2	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Read	VER 3	VER 2	VER 1	VER 0	PNUM 15	PNUM 14	PNUM 13	PNUM 12	PNUM 11	PNUM 10	PNUM 9	PNUM 8	PNUM 7	PNUM 6	PNUM 5	PNUM 4
Write																
Reset	0	0	0	0	0	0	0	1	1	1	1	1	0	1	0	1

Figure 17-4. JTAG Chip Identification (CID) Register

The device identification number for the initial release of the 5685x family is $01F5A01D.

Table 17-4. Device ID Register Bit Assignment

Bit No.	Code Use	56853/854/855/857/858 Values
31–28	Version Number	0000 (For initial version only—these bits may vary)
27–22	Design Center ID	00 0111
21–12	Family and part ID	11 0101 1010
11-1	Manufacturer ID	000 0000 1110
0	IEEE Requirement	Always 1

17.6 Bypass Register (BYPASS)

The JTAG bypass register is a one-bit register used to provide a simple, direct path from the TDI pin to the TDO pin. This is useful in boundary scan applications where many chips are serially connected in a daisy-chain. Individual DSCs, or other devices, can be programmed with the BYPASS instruction so individually they become pass-through devices during testing. This allows testing of a specific chip, while still having all of the chips connected through the JTAG ports.

IR = $6, $7, $FF	
Read/Write	
Reset	0

Figure 17-5. JTAG Bypass Register (JTAGBR)

17.7 JTAG Boundary Scan Register (BSR)

The JTAG Boundary Scan Register (BSR) is configured as described in **Figure 17-6.** This register is enabled via the JTAG Master TAP by issuing the EXTEST, or SAMPLE_PRELOAD instructions enabling the boundary scan registers between TDI and TDO. Boundary Scan Register cell number one is connected to TDO making it the first data bit shifted into TDI. It is the first bit shifted out of TDO when loading and unloading the boundary scan chain. For the most current BSDL files, please refer to www.freescale.com. **Figure 17-6** illustrates the register, while **Table 17-5** provides the contents of the BSR for the 5685x.

IR = $0, $1, $3	337	336	335	334	333	Bits 332 through 5						4	3	2	1	0
Read-Only																

Figure 17-6. Boundary Scan Register (BSR)

Table 17-5. BSR Contents for 5685x

Bit Number	Pin/Bit Name	Pin Type	BSR Cell	853 Pin Number (128 LQFP Package)	854 Pin Number (128 LQFP Package)	855 Pin Number (100 LQFP Package)	857 Pin Number (100 LQFP Package)	858 Pin Number (144 LQFP Package)	858 Pin Number (144 MAPBGA Package)
0	MISO	Input/Output	BC_7	1	1	—	1	1	B2
1		Pull-up	BC_1						
2		Control	BC_2						
3	MOSI	Input/Output	BC_7	2	2	—	2	2	C3
4		Pull-up	BC_1						
5		Control	BC_2						
6	SCK	Input/Output	BC_7	3	3	—	3	3	C2
7		Pull-up	BC_1						
8		Control	BC_2						
9	SS	Input/Output	BC_7	4	4	—	4	4	D2
10		Pull-up	BC_1						
11		Control	BC_2						
12	RD	Input/Output	BC_7	7	7	3	—	8	D3
13		Pull-up	BC_1						
14		Control	BC_2						
15	WR	Input/Output	BC_7	8	8	4	—	9	D4
16		Pull-up	BC_1						
17		Control	BC_2						
18	A0	Input/Output	BC_7	9	9	5	—	10	E5
19		Pull-up	BC_1						
20		Control	BC_2						
21	A1	Input/Output	BC_7	10	10	6	—	11	E4
22		Pull-up	BC_1						
23		Control	BC_2						
24	A2	Input/Output	BC_7	11	11	7	—	12	E3
25		Pull-up	BC_1						
26		Control	BC_2						
27	A3	Input/Output	BC_7	12	12	8	—	13	E2
28		Pull-up	BC_1						
29		Control	BC_2						
30	MDOA	Input/Output	BC_7	15	15	11	10	17	F4
31		Pull-up	BC_1						
32		Control	BC_2						
33	MODB	Input/Output	BC_7	16	16	12	11	18	F3
34		Pull-up	BC_1						
35		Control	BC_2						

Table 17-5. BSR Contents for 5685x (Continued)

Bit Number	Pin/Bit Name	Pin Type	BSR Cell	853 Pin Number (128 LQFP Package)	854 Pin Number (128 LQFP Package)	855 Pin Number (100 LQFP Package)	857 Pin Number (100 LQFP Package)	858 Pin Number (144 LQFP Package)	858 Pin Number (144 MAPBGA Package)
36	MODC	Input/Output	BC_7	17	17	13	12	19	F2
37		Pull-up	BC_1						
38		Control	BC_2						
39	*	Internal	BC_1	—	—	—	—	—	—
40	IRQA	Input	BC_1	20	20	16	15	22	G2
41		Pull-up	BC_1						
42	IRQB	Input	BC_1	21	21	17	16	23	F5
43		Pull-up	BC_1						
44	A4	Input/Output	BC_7	26	26	22	—	29	J2
45		Pull-up	BC_1						
46		Control	BC_2						
47	A5	Input/Output	BC_7	27	27	23	—	30	H3
48		Pull-up	BC_1						
49		Control	BC_2						
50	A6	Input/Output	BC_7	28	28	24	—	31	G4
51		Pull-up	BC_1						
52		Control	BC_2						
53	A7	Input/Output	BC_7	29	29	25	—	32	H4
54		Pull-up	BC_1						
55		Control	BC_2						
56	HD0	Input/Output	BC_7	30	30	—	22	33	J3
57		Pull-up	BC_1						
58		Control	BC_2						
59	HD1	Input/Output	BC_7	31	31	—	23	34	K2
60		Pull-up	BC_1						
61		Control	BC_2						
62	HD2	Input/Output	BC_7	32	32	—	24	35	L2
63		Pull-up	BC_1						
64		Control	BC_2						
65	CLKO	Output	BC_1	33	33	26	26	37	L3
66		Control	BC_1						
67	RSTO	Output	BC_1	34	34	27	27	38	K3
68		Control	BC_1						
69	HD3	Input/Output	BC_7	36	36	—	29	40	J4
70		Pull-up	BC_1						
71		Control	BC_2						

Table 17-5. BSR Contents for 5685x (Continued)

Bit Number	Pin/Bit Name	Pin Type	BSR Cell	853 Pin Number (128 LQFP Package)	854 Pin Number (128 LQFP Package)	855 Pin Number (100 LQFP Package)	857 Pin Number (100 LQFP Package)	858 Pin Number (144 LQFP Package)	858 Pin Number (144 MAPBGA Package)
72	HD4	Input/Output	BC_7	37	37	—	30	41	L4
73		Pull-up	BC_1						
74		Control	BC_2						
75	HD5	Input/Output	BC_7	38	38	—	31	42	J5
76		Pull-up	BC_1						
77		Control	BC_2						
78	HD6	Input/Output	BC_7	39	39	—	32	43	K5
79		Pull-up	BC_1						
80		Control	BC_2						
81	HD7	Input/Output	BC_7	40	40	—	33	44	H5
82		Pull-up	BC_2						
83		Control	BC_2						
84	A8	Input/Output	BC_7	43	43	31	—	48	G5
85		Pull-up	BC_1						
86		Control	BC_2						
87	A9	Input/Output	BC_7	44	44	32	—	49	L5
88		Pull-up	BC_1						
89		Control	BC_2						
90	A10	Input/Output	BC_7	45	45	33	—	50	J6
91		Pull-up	BC_1						
92		Control	BC_2						
93	A11	Input/Output	BC_7	46	46	34	—	51	K6
94		Pull-up	BC_1						
95		Control	BC_2						
96	DE	Input/Output	BC_7	49	49	37	39	55	H6
97		Pull-up	BC_1						
98		Control	BC_2						
99	*	Internal	BC_1	—	—	—	—	—	—
100	A12	Input/Output	BC_7	57	57	45	—	63	J8
101		Pull-up	BC_1						
102		Control	BC_2						
103	A13	Input/Output	BC_7	58	58	46	—	64	K8
104		Pull-up	BC_1						
105		Control	BC_2						
106	A14	Input/Output	BC_7	59	59	47	—	65	L9
107		Pull-up	BC_1						
108		Control	BC_2						

Table 17-5. BSR Contents for 5685x (Continued)

Bit Number	Pin/Bit Name	Pin Type	BSR Cell	853 Pin Number (128 LQFP Package)	854 Pin Number (128 LQFP Package)	855 Pin Number (100 LQFP Package)	857 Pin Number (100 LQFP Package)	858 Pin Number (144 LQFP Package)	858 Pin Number (144 MAPBGA Package)
109	A15	Input/Output	BC_7	60	60	48	—	66	K9
110		Pull-up	BC_1						
111		Control	BC_2						
112	RXD0	Input/Output	BC_7	65	65	51	51	73	L10
113		Pull-up	BC_1						
114		Control	BC_2						
115	TXD0	Input/Output	BC_7	66	66	52	52	74	L11
116		Pull-up	BC_1						
117		Control	BC_2						
118	A16	Input/Output	BC_7	67	67	53	—	75	K10
119		Pull-up	BC_1						
120		Control	BC_2						
121	A17	Input/Output	BC_7	68	68	54	—	76	K11
122		Pull-up	BC_1						
123		Control	BC_2						
124	A18	Input/Output	BC_7	69	69	55	—	77	J9
125		Pull-up	BC_1						
126		Control	BC_2						
127	A19	Input/Output	BC_7	70	70	56	—	78	J10
128		Pull-up	BC_1						
129		Control	BC_2						
130	A20	Input/Output	BC_7	71	71	57	—	79	J11
131		Pull-up	BC_1						
132		Control	BC_2						
133	D0	Input/Output	BC_7	73	73	59	—	81	H7
134		Pull-up	BC_1						
135		Control	BC_2						
136	CS0	Input/Output	BC_7	75	75	61	55	83	H8
137		Pull-up	BC_1						
138		Control	BC_2						
139	CS1	Input/Output	BC_7	76	76	62	56	84	H9
140		Pull-up	BC_1						
141		Control	BC_2						
142	CS2	Input/Output	BC_7	77	77	63	57	85	H11
143		Pull-up	BC_1						
144		Control	BC_2						

Table 17-5. BSR Contents for 5685x (Continued)

Bit Number	Pin/Bit Name	Pin Type	BSR Cell	853 Pin Number (128 LQFP Package)	854 Pin Number (128 LQFP Package)	855 Pin Number (100 LQFP Package)	857 Pin Number (100 LQFP Package)	858 Pin Number (144 LQFP Package)	858 Pin Number (144 MAPBGA Package)
145	CS3	Input/Output	BC_7	78	78	64	58	86	H10
146		Pull-up	BC_1						
147		Control	BC_2						
148	HA0	Input/Output	BC_7	82	82	—	62	90	G10
149		Pull-up	BC_1						
150		Control	BC_2						
151	HA1	Input/Output	BC_7	83	83	—	63	91	G11
152		Pull-up	BC_1						
153		Control	BC_2						
154	HA2	Input/Output	BC_7	84	84	—	64	92	G9
155		Pull-up	BC_1						
156		Control	BC_2						
157	HRW	Input/Output	BC_7	85	85	—	65	93	G8
158		Pull-up	BC_1						
159		Control	BC_2						
160	D1	Input/Output	BC_7	86	86	67	—	94	G7
161		Pull-up	BC_1						
162		Control	BC_2						
163	D2	Input/Output	BC_7	87	87	68	—	95	F9
164		Pull-up	BC_1						
165		Control	BC_2						
166	D3	Input/Output	BC_7	88	88	69	—	96	F10
167		Pull-up	BC_1						
168		Control	BC_2						
169	D4	Input/Output	BC_7	89	89	70	—	97	F11
170		Pull-up	BC_1						
171		Control	BC_2						
172	D5	Input/Output	BC_7	90	90	71	—	98	E10
173		Pull-up	BC_1						
174		Control	BC_2						
175	STD1	Input/Output	BC_7	—	—	—	66	99	E8
176		Pull-up	BC_1						
177		Control	BC_2						
178	SRD1	Input/Output	BC_7	—	—	—	67	100	E11
179		Pull-up	BC_1						
180		Control	BC_2						

Table 17-5. BSR Contents for 5685x (Continued)

Bit Number	Pin/Bit Name	Pin Type	BSR Cell	853 Pin Number (128 LQFP Package)	854 Pin Number (128 LQFP Package)	855 Pin Number (100 LQFP Package)	857 Pin Number (100 LQFP Package)	858 Pin Number (144 LQFP Package)	858 Pin Number (144 MAPBGA Package)
181	SCK1	Input/Output	BC_7	—	—	—	68	101	E9
182		Pull-up	BC_1						
183		Control	BC_2						
184	SC10	Input/Output	BC_7	—	—	—	69	102	D10
185		Pull-up	BC_1						
186		Control	BC_2						
187	SC11	Input/Output	BC_7	—	—	—	70	103	D11
188		Pull-up	BC_1						
189		Control	BC_2						
190	SC12	Input/Output	BC_7	—	—	—	71	104	C11
191		Pull-up	BC_1						
192		Control	BC_2						
193	RXD1	Input/Output	BC_7	94	94	74	74	107	B11
194		Pull-up	BC_1						
195		Control	BC_2						
196	TXD1	Input/Output	BC_7	95	95	75	75	108	C10
197		Pull-up	BC_1						
198		Control	BC_2						
199	TIO3	Input/Output	BC_7	97	97	—	77	110	B10
200		Pull-up	BC_1						
201		Control	BC_2						
202	TIO2	Input/Output	BC_7	98	98	—	78	111	D9
203		Pull-up	BC_1						
204		Control	BC_2						
205	TIO1	Input/Output	BC_7	99	99	—	79	112	C9
206		Pull-up	BC_1						
207		Control	BC_2						
208	TIO0	Input/Output	BC_7	101	101	77	81	114	B9
209		Pull-up	BC_1						
210		Control	BC_2						
211	HDS	Input/Output	BC_7	103	103	—	83	116	C8
212		Pull-up	BC_1						
213		Control	BC_2						
214	HCS	Input/Output	BC_7	104	104	—	84	117	D8
215		Pull-up	BC_1						
216		Control	BC_2						

Table 17-5. BSR Contents for 5685x (Continued)

Bit Number	Pin/Bit Name	Pin Type	BSR Cell	853 Pin Number (128 LQFP Package)	854 Pin Number (128 LQFP Package)	855 Pin Number (100 LQFP Package)	857 Pin Number (100 LQFP Package)	858 Pin Number (144 LQFP Package)	858 Pin Number (144 MAPBGA Package)
217	HREQ	Input/Output	BC_7	105	105	—	85	118	B8
218		Pull-up	BC_1						
219		Control	BC_2						
220	HACK	Input/Output	BC_7	106	106	—	86	119	C7
221		Pull-up	BC_1						
222		Control	BC_2						
223	D6	Input/Output	BC_7	107	107	79	—	120	D7
224		Pull-up	BC_1						
225		Control	BC_2						
226	D7	Input/Output	BC_7	108	108	80	—	121	B7
227		Pull-up	BC_1						
228		Control	BC_2						
229	D8	Input/Output	BC_7	109	109	81	—	122	E7
230		Pull-up	BC_1						
231		Control	BC_2						
232	D9	Input/Output	BC_7	110	110	82	—	123	F8
233		Pull-up	BC_1						
234		Control	BC_2						
235	D10	Input/Output	BC_7	111	111	83	—	124	F7
236		Pull-up	BC_1						
237		Control	BC_2						
238	STD0	Input/Output	BC_7	116	116	88	92	131	B6
239		Pull-up	BC_1						
240		Control	BC_2						
241	SRD0	Input/Output	BC_7	117	117	89	93	132	C6
242		Pull-up	BC_1						
243		Control	BC_2						
244	SCK0	Input/Output	BC_7	118	118	90	94	133	C5
245		Pull-up	BC_1						
246		Control	BC_2						
247	SC00	Input/Output	BC_7	119	119	91	95	134	D6
248		Pull-up	BC_1						
249		Control	BC_2						
250	SC01	Input/Output	BC_7	120	120	92	96	135	B5
251		Pull-up	BC_1						
252		Control	BC_2						

Table 17-5. BSR Contents for 5685x (Continued)

Bit Number	Pin/Bit Name	Pin Type	BSR Cell	853 Pin Number (128 LQFP Package)	854 Pin Number (128 LQFP Package)	855 Pin Number (100 LQFP Package)	857 Pin Number (100 LQFP Package)	858 Pin Number (144 LQFP Package)	858 Pin Number (144 MAPBGA Package)
253	SC02	Input/Output	BC_7	121	121	93	97	136	E6
254		Pull-up	BC_1						
255		Control	BC_2						
256	D11	Input/Output	BC_7	122	122	94	—	137	D5
257		Pull-up	BC_1						
258		Control	BC_2						
259	D12	Input/Output	BC_7	123	123	95	—	138	B4
260		Pull-up	BC_1						
261		Control	BC_2						
262	D13	Input/Output	BC_7	126	126	98	—	142	C4
263		Pull-up	BC_1						
264		Control	BC_2						
265	D14	Input/Output	BC_7	127	127	99	—	143	F6
266		Pull-up	BC_1						
267		Control	BC_2						
268	D15	Input/Output	BC_7	128	128	100	—	144	B3
269		Pull-up	BC_1						
270		Control	BC_2						

* Test mode

17.8 TAP Controller

The TAP Controller is a synchronous 16-bit finite state machine illustrated in **Figure 17-7.** It responds to changes at the TMS and TCK pins. Transitions from one state to another will occur on the rising edge of TCK. The value shown adjacent to each state transition represents the signal present on TMS at the time of a rising edge of TCK.

The TDO pin will remain in the *high* impedance state except during the Shift-DR and Shift-IR TAP Controller states. In these controller states, TDO will update on the falling edge of TCK. TDI is sampled on the rising edge of TCK.

The TAP Controller will execute the last instruction decoded until a new instruction is entered at the Update-IR state, or Test-Logic-Reset is entered.

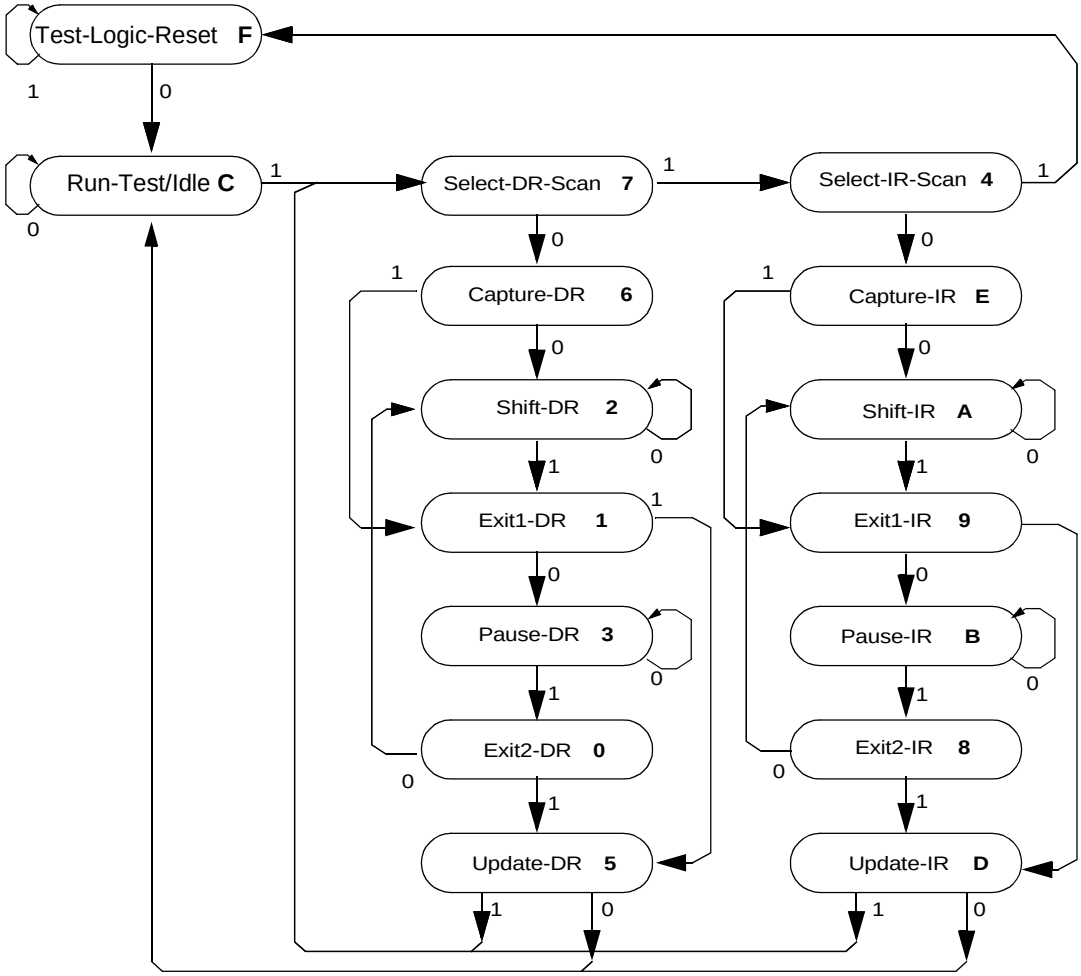


Figure 17-7. TAP Controller State Diagram

The TAP Controller will execute the last instruction decoded until a new instruction is entered at the Update-IR state, or Test-Logic-Reset is entered.

There are two paths through the 16-state machine. The shift-IR-scan path captures and loads JTAG instructions into the JTAGIR. The shift-DR-scan path captures and loads data into the other JTAG registers. The TAP Controller executes the last instruction decoded until a new instruction is entered at the update-IR state, or until the test-logic-reset state is entered. When using the JTAG port to access EOnCE module registers, follow these four steps:

1. Enable the TLM by shifting the TLM_SEL instruction into the JTAGIR.
2. When selected, the TLM must enable the 56800E TAP by shifting in the appropriate value into the TLM register.
3. When the 56800E TAP is selected, the EOnCE module is selected by shifting the ENABLE_EOnCE instruction.

4. The EOnCE module registers and commands are read and written through the JTAG pins using the shift-DR-scan path.

Asserting the JTAG's $\overline{\text{TRST}}$ pin asynchronously forces the JTAG state machine into the test-logic-reset state.

17.8.1 Operation

All state transitions of the TAP Controller occur based on the value of TMS at the time of a rising edge of TCK. Actions of the instructions occur on the falling edge of TCK in each controller state illustrated in **Figure 17-1.**

17.8.1.1 Test Logic Reset (pstate = F)

During Test-Logic-Reset all JTAG test logic is disabled so the chip can operate in a normal mode. This is achieved by initializing the Instruction Register (IR) with the IDCODE instruction. By holding TMS *high* for five rising edges of TCK, the device will always remain in Test-Logic-Reset no matter what state the TAP Controller was in previously.

17.8.1.2 Run-Test-Idle (pstate = C)

Run-Test-Idle is a controller state between scan operations. EOnCE entered, the controller will remain in the Run-Test-Idle mode as long as TMS is held *low*. When TMS is *high* and a rising edge of TCK occurs, the controller moves to the Select-DR state.

17.8.1.3 Select Data Register (pstate = 7)

The Select-DR state is a temporary state. In this state, all Test Data registers selected by the current instruction retains their previous states. If TMS is held *low* and a rising edge of TCK occurs when the controller is in this state, the controller moves into the Capture-DR state and a scan sequence for the selected Test Data register is initiated. If TMS is held *high* and a rising edge of TCK occurs, the controller moves to the Select-IR state.

17.8.1.4 Select Instruction Register (pstate = 4)

The Select-IR state is a temporary state. In this state, all Test Data registers selected by the current instruction retain their previous states. If TMS is held *low* and a rising edge of TCK occurs when the controller is in this state, the controller moves into the Capture-IR state and a scan sequence for the instruction register is initiated. If TMS is held *high* and a rising edge of TCK occurs, the controller moves to the Test-Logic-Reset state.

17.8.1.5 Capture Data Register (pstate = 6)

In this controller state, data may be parallel loaded into test registers selected by the current instruction on the rising edge of TCK. If a test data register selected by the current instruction does not have a parallel input, the register retains its previous value.

17.8.1.6 Shift Data Register (pstate = 2)

In this controller state, the Test Data register is connected between TDI and TDO. This data is then shifted one stage towards its serial output on each rising edge of TCK. The TAP Controller will remain in this state while TMS is held at a *low*. When a one is applied to TMS and a positive edge of TCK occurs the controller will move to the Exit1-DR state.

17.8.1.7 Exit1 Data Register (pstate = 1)

This is a temporary controller state. If TMS is held *high*, and a rising edge is applied to TCK while in this state causes the controller to advance to the Update-DR state. This terminates the scanning process.

17.8.1.8 Pause Data Register (pstate = 3)

This controller state allows shifting of the Test Data register in the serial path between TDI and TDO to be temporarily halted. All test data registers selected by the current instruction retain their previous state unchanged. The controller remains in this state while TMS is held *low*. When TMS goes *high* and a rising edge is applied to TCK, the controller advances to the Exit2-DR state.

17.8.1.9 Exit2 Data Register (pstate = 0)

This is a temporary controller state. If TMS is held *high*, and a rising edge is applied to TCK while it is in this state, the scanning process terminates and the TAP Controller advances to the Update-DR state. If TMS is held *low* and a rising edge of TCK occurs, the controller advances to the Shift-DR state.

17.8.1.10 Update Data Register (pstate = 5)

All boundary scan register contain a two stage data register. It isolates the shifting and capturing of data on the peripheral from what is applied to internal logic during scan mode. This register is the second stage, or parallel output, and it is used to apply a stimulus to internal logic. Data is latched on the parallel output of these Test Data registers from the Shift register path on the falling edge of TCK in the Update-DR state. On a rising edge of TCK, the controller advances to the Select_DR state if TMS is held *high* or the Run-Test-Idle state If TMS is held *low*.

17.8.1.11 Capture Instruction Register (pstate = E)

When the TAP Controller is in this state and a rising edge of TCK occurs, the controller advances to the Exit1-IR state if TMS is held at a one or the Shift-IR state if TMS is held at a zero.

17.8.1.12 Shift Instruction Register (pstate = A)

In this controller state, the Shift register contained in the Instruction Register (IR) is connected between TDI and TDO and shifts data one stage towards it's serial output on each rising edge of TCK. When the TAP Controller is in this state and a rising edge of TCK occurs, the controller advances to the Exit1-IR state if TMS is held at a one or remains in the Shift-IR state if TMS is held at a zero.

17.8.1.13 Exit1 Instruction Register (pstate = 9)

This is a temporary controller state. If TMS is held *high*, and a rising edge is applied to TCK while in this state causes the controller to advance to the Update-IR state. This terminates the scanning process. If TMS is held *low* and a rising edge of TCK occurs the controller advances to the Pause-IR state.

17.8.1.14 Pause Instruction Register (pstate = B)

This controller state allows shifting of the Instruction Register (IR) in the serial path between TDI and TDO to be temporarily halted. All Test Data registers selected by the current instruction retain their previous state unchanged. The controller remains in this state while TMS is held *low*. When TMS goes *high* and a rising edge is applied to TCK, the controller advances to the Exit2-IR state.

17.8.1.15 Exit2 Instruction Register (pstate = 8)

This is a temporary controller state. If TMS is held *high*, and a rising edge is applied to TCK while in this state, the scanning process terminates and the TAP Controller advances to the Update-IR state. If TMS is held *low* and a rising edge of TCK occurs, the controller advances to the Shift-IR state.

17.8.1.16 Update Instruction Register (pstate = D)

During this state, instruction shifted into the Instruction Register (IR) is latched from the Shift register path on the falling edge of TCK and into the instruction latch. It becomes the current instruction. On a rising edge of TCK, the controller advances to the Select_IR state if TMS is held *high* or the Run-Test-Idle state If TMS is held *low*.

17.9 5685x Restrictions

The control afforded by the output enable signals using the BSR and the EXTEST instruction requires a compatible circuit board test environment to avoid any device-destructive configurations. *Avoid situations when the 5685x output drivers are enabled into actively driven networks.*

During power-up, the $\overline{\text{TRST}}$ pin must be externally asserted to force the TAP Controller into this state. After power-up is concluded, TMS must be sampled as a Logic 1 for five consecutive TCK rising edges. If TMS either remains unconnected or is connected to V_{DD}, then the TAP Controller cannot leave the test-logic-reset state, regardless of the state of TCK.

5685x features a low-power Stop mode invoked using the stop instruction. JTAG interaction with low-power Stop mode is as follows:

1. The TAP Controller must be in the test-logic-reset state to either enter or remain in Stop mode. Leaving the TAP Controller test-logic-reset state negates the ability to achieve low-power, but does not otherwise affect device functionality.
2. The TCK input is not blocked in low-power Stop mode. To consume minimal power, the TCK input should be tied to ground only.
3. The TMS and TDI pins include On-Chip Pull-Up resistors. In low-power Stop mode, these two pins should remain either unconnected or connected to V_{DD} to achieve minimal power consumption.

Because all 5685x clocks are disabled during Stop state, the JTAG interface provides the means of polling the device status, sampled in the Capture-IR state.

Appendix A
Glossary

A.1 Glossary

This glossary is intended to reduce potential confusion caused by the use of many acronyms and abbreviations throughout this manual.

ACIM	A/C Induction Motors
A/D	Analog-to-Digital
ADC	Analog to Digital Converter
ADCR	ADC Control Registe
ADDR	Address
ADHLMT	ADC High Limit Registers
ADLLMT	ADC Low Limit Registers
ADLST	ADC Channel List Registers
ADLSTAT	ADC Limit Status Register
ADM	Application Development Module
ADOFS	ADC Offset Registers
ADR PD	Address Bus Pull-up Disable
ADRSLT	ADC Result Registers
ADSDIS	ADC Sample Disable Register
ADSTAT	ADC Status Register
ADZCC	ADC Zero Crossing Control Register
ADZCSTAT	ADC Zero Crossing Status Register
AGU	Address Generation Unit
ALU	Arithmetic Logic Unit
API	Application Program Interface
Barrel Shifter	Part of the ALU that allows single cycle shifting and rotating of data word
BCR	Bus Control Register
BDC	Brush DC Motor
BE	Breakpoint Enable
BFIU	Boot Flash Interface Unit
BFLASH	Boot Flash
BK	Breakpoint Configuration Bit
BLDC	Brushless DC Motor
BLKSZ	Base Address and Block Size Register in the EMI peripherial

BOTNEG	Bottom-side PWM Polarity Bit
BS	Breakpoint Selection
BSDL	Boundary Scan Description Language
BSR	Boundary Scan Register
CAN	Controller Area Network
CC	Condition Codes
CAP	Capture
CDBR	Core Data Bus Read
CDBW	Core Data Bus Write
CEN	COP Enable Bit
CFG	Config
CGDB	Core Global Data Bus
CGM	Clock Generator Module
CGMDB	Clock Generator Module Divide-By Register in the OCCS Module
CGMTOD	Clock Generator Module Time of Day Register in the OCCS Module
CGMTST	Clock Generator Module Test Register in the OCCS Module
CHCNF	Channel Configure
CID	Chip Identification Register
CKDIVISOR	Clock Divisor
CLKO	Clock Output pin
CLKOSEL	CLKO Select
CLKOSR	Clock Select Register
CMOS	Complementary metal oxide semiconductor. (A form of digital logic that is characterized by low power consumption, wide power suppply range, and high noise immunity.)
CMP	Compare
CNT	Count
CNTR	Counter
Codec	Coder/Decoder
COP	Computer Operating Properly
COP/RTI	Computer Operating Properly/Real Time Interface
COPCTL	COP Control
COPDIS	COP Timer Disable
COPR	COP Reset

COPSRV	COP Service
COPTO	COP Time Out
CP	Charge Pump
CPHA	Clock Phase
CPOL	Clock Polarity
CPU	Central Processing Unit
CRC	Cyclic Redundancy Code
CS	Chip Select
CSEN	Cop Stop Enable
CSOR	Chip Select Option Register in the EMI peripheral
CTRL	Control
CTRL PD	Control signal Pull-up Disable
CVR	Command Vector Regsiter
CWEN	COP Wait Enable Bit
CWP	COP Write Protect
DAC	Digital to Analog Converter
DAT	Data/Address Select
DATA ALU	Data Address Limit
DATA PD	Data bus I/O Pull-up Disable
DC	Down Counter programmable divide by *n* counter
DDA	Analog Power
DDR	Data Direction Register
DEC	Quadrature Decoder Module
DEE	Dumb Erase Enable
DFIU	Data Flash Interface Unit
DFLASH	Data Flash
DIE	Watchdog Time-Out Interrupt Enable
DIRQ	Watchdog Time-Out Interrupt Request
DM	Data Memory
DMA	Direct Memory Access
DMADR	Data Memory Address
DMW	Data Memory Write

DPE	Dumb Programming Enable
DR	Data Register
DRV	Drive Control Bit
DSC	Digital Signal Controller
DSO	Data Shift Order
DSP	Digital Signal Processor
EDG	Edge-Aligned or Center-Aligned PWMs
EE	Erase Enable
EEOF	Enable External OFLAG Force
EM	Event Modifier
EMI	External Memory Interface
EN	Enable3
ENA	Enables (TAP TLM)
ENCR	Encoder Control Register
EOSI	End of Scan Interrupt
EOSIE	End of Scan Interrupt Enable
ERASE	Erase Cycle
ERRIE	Error Interrupt Enable
EX	External X Memory
EXTBOOT	External Boot
EXTR	External Reset
FAULT	Fault Input to PWM
FE	Framing Error Flag
FLAGx	FAULTx Pin Flag
FH	FIFO Halt
FIEx	Faultx Pin Interrupt Enable
FSM	Finite State Machine
FIR	Filter Interval Register
FLOCI	Force Loss of Clock
FLOLI	Force Loss of Lock
FMODEx	FAULTx Pin Clearing Mode
FOSC	Oscillator Frequency

FPINx	FAULTx Pin
FREF	Reference Frequency
FTACKx	FAULTx Pin Acknowledge
GPIO	General Purpose Input/Output
GPR	Group Priority Register
Harvard Architecture	A microprocessor architecture using separate buses for program and data. This is data is typically used on DSPs to optimise the data throughput.
HACK	Host Acknowledge Input Pin
HBO	Hardware Breakpoint Occurrence
HC	Host Command Bit
HCIE	Host Command Interrupt Enable Bit
HCP	Host Command Pending Bit
HCR	Host Interface Control Register
HDDS	Host Dual Data Strobe Bit
HDMA	Host DMA Status Bit
HF0	Host Flag 0 Bit (general-purpose flag)
HF1	Host Flag 1 Bit (general-purpose flag)
HF2	Host Flag 2 Bit (general-purpose flag)
HF3	Host Flag 3 Bit (general-purpose flag)
HLEND	Host Little Endian Bit
HLMTI	High Limit Interrupt Bit
HLMTIE	High Limit Interrupt Enable Bit
HM0	Host Mode Control 0 Bit
HM1	Host Mode Control 1 Bit
HOLD	Hold Register
HOME	Home Switch Input
HRDF	Host Status Receive Data Full Bit
HREQ	Host Request Output Bit
HRIE	Host Receive Interrupt Enable Bit
HRMS	Host Request Mode Select Bit
HRRQ	Host Receive Request Bit
HRX	Host Interface Data Register

HSR	Host Interface Status Register
HTDE	Host Transmit Data Empty Bit
HTIE	Host Transmit Interrupt Enable Bit
HTRQ	Host Transmit Request Bit
HTX	Host Transmit Data Register
HV	Host Vector Bits
IA	Interrupt Assert
IC	Integrated Circuit
ICR	Interface Control Register
IE	Interrupt Enable
IEE	Intelligent Erase Enable
IEF	Input Edge Flag
IEFIE	Input Edge Flag Interrupt Enable
IENR	Interrupt Enable Register
IES	Interrupt Edge Sensitive
IFREN	Information Block Enable
IMR	Input Monitor Register
INDEP	Independent or Complimentary Pair Operation
INDEX	Index Input
INIT	Initialize Bit
INPUT	External Input Signal
INV	Invert
I/O	Input/Output
IP	Interrupt Pending
IPBus	Intellectual Properties Bus
IPE	Intelligent Program Enable
IPOL	Current Polarity
IPOLR	Interrupt Polarity Register
IPBBA	Interrupt Properties Bus Bridge Address
IPBB	Interrupt Pending Bus Bridge
IPR	Interrupt Pending Register (in GPIO)
IPR	Interrupt Priority Register (in the Core)

IPS	Input Polarity Select
IRQ	Interrupt Request
IS	Interrupt Source
ISC	In Select Control (TAP TLM)
ISR	Interface Status Register
IVR	Interrupt Vector Regsiter
ITCN	Interrupt Controller
JTAG	Joint Test Action Group
JTAGBR	JTAG Bypass Register
JTAGIR	JTAG Instruction Register
LC	Link Controls
LCD	Liquid Crystal Display
LCK	Loss of Lock
LDOK	Load OKay
LF	Loop Filter
LIR	Lower Initialization Register
LLMTI	Low Limit Interrupt
LLMTIE	Low Limit Interrupt Enable
LOAD	Load Register
LOCI	Loss of Clock
LOCIE	Los of Clock Interrupt Enable
LOLI	PLL Lock of Lock Interrupt
LOOP	Loop Select Bit
LPOS	Lower Position Counter Register
LPOSH	Lower Position Hold Register
LSB	Least Significant Bit
LSH_ID	Most Significant Half of JTAG_ID
LVD	Low Voltage Detect
LVIE	Low Voltage Interrupt Enable
LVIS	Low Voltage Interrupt Source
M	Mode
MA	Mode A

MAC	Multiply and Accumulate
MAS	Mass Cycle Erase
MB	Mode B
MCU	Microcontroller Unit -
MHz	Megahertz
MIPS	Million Instructions Per Second
MISO	Master In/Slave Out
MODF	Mode Fault Error
MODFEN	Mode Fault Enable
MOSI	Master Out/Slave In
MPIO	Multi-Purpose Input/Output (A, B, C, D, E or F)
MSB	Most Significant Bit
MSCAN	Motorola Scalable Controller Area Network
MSH_ID	Most Significant Half of JTAG ID
MSTR	Master Mode
MUX	Multiplexer
NF	Noise Flag
NL	Nested Looping
NOR	An inversion of the logical OR function
NVSTR	Non-volatile Store Cycle Definition
OBAR	OnCE Breakpoint Address Register
OBCTL	OnCE Breakpoint Control Register
OBMSK	OnCE Breakpont Mask Register
OCMDR	OnCE Command Register
OCCS	On-Chip Clock Synthesis
OCNTR	OnCE Count Register
OCR	OnCE Control Register
ODEC	OnCE Decoder
OEN	Output Enable
OMAC	OnCE Memory Address Comparator
OMAL	OnCE Memory Address Latch
OMR	Operating Mode Register

OnCE	On-Chip Emulation (unit)
OPABDR	OnCE Program Address Bus Decode Register
OPABER	OnCE Program Address Bus Execute Register
OPABFR	OnCE Program Address Bus Fetch Register
OPDBR	OnCE Program Data Bus Register
OPFIFO	OnCE PAB Change of Flow
OPGDBR	OnCE Program Global Data Bus Register
OPS	Output Polarity Select
OR	Overrun
OSHR	OnCE Shift Register
OSR	OnCE Status Register
OVRF	Overflow
PAB	Program Address Bus
PD	Permanent STOP/WAIT Disable
PDB	Program Data Bus
PE	Program Enable
PE	Parity Enable Bit
PER	Peripheral Enable Register
PF	Parity Error Flag
PFD	Phase Frequency Detector
PFIU	Program Flash Interface Unit
PFLASH	Program Flash
PGDB	Peripheral Global Data Bus
PLL	Phase Locked Loop Module
PLLCID	PLL Clock In Divide
PLLCOD	PLL Clock Out Divide
PLLDB	PLL Divide-by
PLLCR	PLL Control Register
PLLPDN	PLL Power Down
PLLSR	PLL Status Register
PLR	Priority Level Register
PMCCR	PWM Channel Control Register

PMCFG	PWM Configuration Register
PMCNT	PWM Counter Register
PMCTL	PWM Control Register
PMDEADTM	PWM Deadtime Register
PMDISMAP	PWM Disable Mapping Registers
PMFCTL	PWM Fault Control Register
PMFSA	PWM Fault Status Acknowledge
PMOUT	PWM Output Control Register
PMPORT	PWM Port Register
POL	Polarity
POR	Power on Reset
PRAM	Program RAM
PROG	Program Cycle
PSR	Processor Status Register
PT	Parity Type
PTM	Peripheral Test Mode
PUR	Pull-up Enable Register
PWD	Power Down Mode
PWM	Pulse Width Modulator
PWMEN	PWM Enable
PWMF	PWM Reload Flag
PWMRIE	PWM Reload Interrupt Enable
PWMVAL	PWM Value Registers
QE	Quadrature Encoder
QDN	Quadrature Decoder Negative Signal
RAF	Receiver Active Flag
RAM	Random Access Memory
RDRF	Receive Data Register Full
RE	Receiver Enable
REIE	Receive Error Interrupt Enable
REV	Revolution Counter Register
REVH	Revolution Hold Register

RDMAEN	Receive DMA Enable Bit
RIDLE	Receiver Idle Line
RIE	Receiver Full Interrupt Enable
ROM	Read Only Memory
RPD	Re-programmable STOP/WAIT Disable
RREQ	Receive Request Bit
RSRC	Receiver Source Bit
RWU	Receiver Wake up
RXDF	Receive Data Register Full Bit
RXH	Receive Byte High Register
RXL	Receive Byte Low Register
SA	Saturation
Sample	A word or time-slot of data to be transferred in a frame
SBK	Send Break
SBO	Software Breakpoint Occurrence
SBR	SCI Baud Rate
SCI	Serial Communications Interface3
SCIBR	SCI Baud Rate Register
SCICR	SCI Control Register
SCIDR	SCI Data Register
SCISR	SCI Status Register
SCLK	Serial Clock
SCR	Status and Control
SD	Stop Delay
SDK	Software Development Kit
SEL	Selects (TAP TLM)
SEXT	Sign Extend
SIM	System Integration Module
SMODE	Scan Mode
SPDRR	SPI Data Receive Register
SPDSR	SPI Data Size Register
SPDTR	SPI Data Transmit Register

SP	SPI Enable
SPI	Serial Peripheral Interface
SPMSTR	SPI Master
SPRF	SPI Receiver Full
SPRIE	SPI Receiver Interrupt Enable
SPSCR	SPI Status Control Register
SPTE	SPI Transmitter Empty
SPTIE	SPI Transmit Interrupt Enable
SR	Status Register
SRM	Switched Reluctance Motor
$\overline{\textbf{SS}}$	Slave Select
SSI	Synchronous Serial Interface
SWAI	Stop in Wait Mode
SYS_CNTL	System Control Register
SYS_STS	System Status Register
TAP	Test Access Port
TCSR	Text Control and Status Register
TCE	Test Counter Enable
TCF	Timer Compare Flag
TCFIE	Timer Compare Flag Interrupt Enable
TCK	TAP Clock
TDI	TAP Data In
TDO	TAP Data Out
TDMAEN	Transmit DMA Enable Bit
TDRE	Transmit Date Register Empty
TE	Transmitter Enable
TEIE	Transmitter Empty Interrupt Enable
TEN	Test Mode Enable
TERASEL	Terase Limit
TESTR	Test Register
TFDBK	Test Feedback Clock
TFREF	Test Reference Frequency Clock

TIDLE	Transmitter Idle
TIIE	Transmitter Idle Interrupt Enable
Time-Slot	A frame divided into time-slots, allowing for the transfer of a word of data
TIRQ	Test Interrupt Request Register
TISR	Test Interrupt Source Register
TM	Test Mode bit
TMEL	Time Limit
TMODE	Test Mode bit
TMR	Quadrature Timer
TMR PD	Timer I/O Pull-up Disable
TNVHL	TNVH Limit
TNVSL	TNVS Limit
TO	Trace Occurrence
TOD	Time of Day Module
TOF	Timer Overflow Flag
TOFIE	Timer Oerflow Flag Interrupt Enable
TOPNEG	Top-side PWM Polarity Bit
TPROGL	Tprog Limit
TPGSL	TPGS Limit
TRCVL	TRCV Limit
TRDY	Transmit Ready Flag Bit
TREQ	Transmit Reuest Enable Bit
TSTREG	Test Register
TXDE	Transmit Data Register Empty Bit
UIR	Upper Initialization Register
UPOS	Upper Position Hold Register
UPOSH	Upper Position Hold Register
VAB	Vector Address Bus
VBA	Vector Base Address are pins on the 56800E core
VCO	Voltage Controlled Oscillator
V_{DD}	Voltage Digital Drain (Power)
V_{DDA}	Analog Power

VEL	Velocity Counter Register
VELH	Velocity Hold Register
VLMODE	Value Register Load Mode
VREF	Voltage Reference
VRM	Variable Reluctance Motor
V_{SS}	Ground
V_{SSA}	Analog Ground
WAKE	Wake up Condition
WDE	Watchdog Enable
WP	Write Protect
WSPM	Wait State P Memory
WSX	Wait State Data Memory
WTR	Watchdog Timeout Register
WWW	World Wide Web
XDB2	X Data Bus
XE	X Address Enable
XIE	Index Pulse Interrupt Enable
XIRQ	Index Pulse Interrupt Request
XNE	Use Negative Edge of Index Pulse
XRAM	Data RAM
YE	Y Address Enable
ZCI	Zero Crossing Interrupt
ZCIE	Zero Crossing Interrupt Enable
ZCS	Zero Crossing Status
ZSRC	Zclock Source

Appendix B
Programmer's Sheets

B.1 Introduction

The following pages provide a set of reference tables and programming sheets intended to simplify programming the 5685x. The programming sheets provide room to add the value of each bit and the hexadecimal value for each register. These pages may be photocopied.

For complete instruction set details, please refer to Chapter 4 of the *DSP56800E Reference Manual* (DSP56800ERM).

B.2 Programmers' Sheets

Each described instruction contains notations used to abbreviate certain operands and operations described in **Table B-1.** Programmers' sheets are arranged to correspond with the chapters in this document. The same table lists programmers' sheets by module, the registers in each module, and the appendix pages where the programmers' sheets are located.

Note: Reserved bits (■) should always be set to 0 unless otherwise stated.

Table B-1. List of Programmer's Sheets

Register Type	Register	Page
SYSTEM INTEGRATION MODULE (SIM)	**SYS_BASE = $1FFF08**	
SIM Control Register	(SCR)	B-8
SIM Control Data Registers 1-2	(SCD1-2)	B-9
EXTERNAL MEMORY INTERFACE (EMI)	**EMI_BASE = $1FFE40**	
Base Address and Block Size Register	(CSBAR)	B-10
Chip Select Option Register	(CSOR)	B-11
Bus Control Register	(BCR)	B-13
ON-CHIP CLOCK SYNTHESIS (OCCS)	**CGM_BASE = $1FFFF10**	
CGM Control Register	(CGMCR)	B-14, B-15
CGM Divide-By Register	(CGMDB)	B-16
CGM Time of Day Register	(CGMTOD)	B-17
COMPUTER OPERATING PROPERLY (COP)	**COP_BASE = $1FFFD0**	
COP Control Register	(COPCTL)	**B-18**
COP Timeout Register	(COPTO)	**B-19**
COP Counter Register	(COPCTR)	**B-20**

Table B-1. List of Programmer's Sheets (Continued)

Register Type	Register	Page
INTERRUPT CONTROL (ITCN)	**ITCN_BASE = $1FFF20**	
Interrupt Priority Register 0	(IPR0)	B-21
Interrupt Priority Register 1	(IPR1)	B-22
Interrupt Priority Register 2	(IPR2)	B-23
Interrupt Priority Register 3	(IPR3)	B-25
Interrupt Priority Register 4	(IPR4)	B-27
Interrupt Priority Register 5	(IPR5)	**B-29**, **B-30**
Interrupt Priority Register 6	(IPR6)	B-31, **B-32**
Interrupt Priority Register 7	(IPR7)	B-33, B-34
Vector Base Address Register	(VBA)	B-37
Fast Interrupt Match Register 0	(FIM0)	B-38
Fast Interrupt Match Register 1	(FIM1)	B-39
Fast Interrupt Vector Address Low 0 and High 0	(FIVAL0 and FIVAH0)	B-40
Fast Interrupt Vector Address Low 1 and High 1	(FIVAL1 and FIVAH1)	B-41
Interrupt Request Pending Register 0-3	(IRQP0-3)	B-42
Interrupt Control Register	(ICTL)	B-43, **B-44**
DIRECT MEMORY ACCESS (DMA)	**DMA0_BASE = $1FFEC0**	
	DMA1_BASE = $1FFEC8	
	DMA2_BASE = $1FFED0	
	DMA3_BASE = $1FFED8	
	DMA4_BASE = $1FFEE0	
	DMA5_BASE = $1FFEE8	
Source Address High Register	(DMASAH)	B-45
Source Address Low Register	(DMASAL)	B-46
Destination Address High Register	(DMADAH)	B-47
Destination Address Low Register	(DMADAL)	B-48
Transfer Count Register	(DMACNT)	B-49
Circular Queue Size Register	(DMACQS)	B-50
Transfer Control Register	(DMATC)	B-51, **B-52**
SERIAL COMMUNICATION INTERFACE (SCI0-1)	**SCI0_BASE = $1FFFE0**	
	SCI1_BASE = $1FFDF8	
Baud Register	(SCIBR)	B-53
Control Register	(SCICR)	B-54, B-55, **B-56**
Control Register 2	(SCICR2)	B-57
Status Register	(SCISR)	B-58, B-59, B-60
Data Register	(SCIDR)	B-61

Table B-1. List of Programmer's Sheets (Continued)

Register Type	Register	Page
SERIAL PERIPHERAL INTERFACE (SPI)	**SPI_BASE = $1FFFE8**	
Status and Control Register	(SPSCR)	B-62, B-63, B-64
Data Size and Control Register	(SPDSCR)	B-65
Data Receive Register	(SPDRR)	B-66
Data Transmit Register	(SPDTR)	B-66

ENHANCED SYNCHRONOUS SERIAL INTERFACE (ESSI)	**ESSI0_BASE = $1FFE20**	
	ESSI1_BASE = $1FFE00	
Transmit Data Register 0-2	(STX0-2)	B-68
Receive Data Register	(SRX)	**B-69**
Status Register	(SSR)	**B-70**, B-71
Control Register 2	(SCR2)	B-72, B-73
Control Register 3	(SCR3)	B-74, B-75
Control Register 4	(SCSR4)	B-76, B-77
Transmit and Receive Control Registers	(STXCR, SRXCR)	B-78
Time Slot Register	(STSR)	B-79
FIFO Control/Status Register	(SFCSR)	B-80, B-81
Transmit Slot Mask Registers	(TSMA, TSMB)	B-82
Receive Slot Mask Registers	(RSMA, RSMB0	B-83

QUAD TIMER (TMR)	**TMR_BASE = $1FFE80**	
Control Register	(CTL)	**B-90**, **B-91**, **B-92**
Status/Control Register	(SCR)	B-93, **B-94**
Compare Register 1	(CMP1)	B-84
Compare Register 2	(CMP2)	**B-85**
Capture Register	(CAP)	**B-86**
Load Register	(LOAD)	**B-87**
Hold Register	(HOLD)	**B-88**
Counter Register	(CNTR)	**B-89**

TIME OF DAY (TOD)	**TOD_BASE = $1FFFC0**	
TOD Control/Status Register	(TODCS)	B-95
Clock Scaler	(TODSCL)	B-96
Seconds Register	(TODSEC)	B-97
Seconds Alarm Register	(TODSAL)	B-98
Minutes Register	(TODMIN)	B-99
Minutes Alarm Register	(TODMAL)	B-100
Hours Register	(TODHR)	B-101

Table B-1. List of Programmer's Sheets (Continued)

Register Type	Register	Page
Hours Alarm Register	(TODHAL)	B-102
Days Register	TODDAY)	B-103
Days Alarm Register	(TODDAL)	B-104

GENERAL PURPOSE IN/OUT (GPIO)	**GPIO A_BASE = $1FFE60**	
	GPIO B_BASE = $1FFE64	
	GPIO C_BASE = $1FFE68	
	GPIO D_BASE = $1FFE6C	
	GPIO E_BASE = $1FFE70	
	GPIO F_BASE = $1FFE74	
	GPIO G_BASE = $1FFE78	
	GPIO H_BASE = $1FFE7C	
Port A Peripheral Enable Register	(MPA_PER)	B-105
Port B Peripheral Enable Register	(MPB_PER)	B-106
Port C Peripheral Enable Register	(MPC_PER)	B-107
Port D Peripheral Enable Register	(MPD_PER)	B-108
Port E Peripheral Enable Register	(MPE_PER)	B-109
Port F Peripheral Enable Register	(MPF_PER)	B-110
Port G Peripheral Enable Register	(MPG_PER)	B-111
Port H Peripheral Enable Register	(MPH_PER)	B-112
Port A Data Direction Register	(MPA_DDR)	**B-113**
Port B Data Direction Register	(MPB_DDR)	B-114
Port C Data Direction Register	(MPC_DDR)	B-115
Port D Data Direction Register	(MPD_DDR)	B-116
Port E Data Direction Register	(MPE_DDR)	**B-117**
Port F Data Direction Register	(MPF_DDR)	B-118
Port G Data Direction Register	(MPG_DDR)	B-119
Port H Data Direction Register	(MPH_DDR)	B-120
Port A Data Register	(MPA_DR)	B-121
Port B Data Register	(MPB_DR)	B-122
Port C Data Register	(MPC_DR)	B-123
Port D Data Register	(MPD_DR)	**B-124**
Port E Data Register	(MPE_DR)	**B-125**
Port F Data Register	(MPF_DR)	**B-126**
Port G Data Register	(MPG_DR)	B-127
Port H Data Register	(MPH_DR)	B-128
Port A Pull-Up Enable Register	(MPA_PUR)	B-129
Port B Pull-Up Enable Register	(MPB_PUR)	B-130
Port C Pull-Up Enable Register	(MPC_PUR)	B-131
Port D Pull-Up Enable Register	(MPD_PUR)	**B-132**
Port E Pull-Up Enable Register	(MPE_PUR)	**B-133**
Port F Pull-Up Enable Register	(MPF_PUR)	**B-134**

Table B-1. List of Programmer's Sheets (Continued)

Register Type	Register	Page
Port G Pull-Up Enable Register	(MPG_PUR)	**B-135**
Port H Pull-Up Enable Register	(MPH_PUR)	B-136

HOST INTERFACE 8	**HI8_BASE = $1FFFD8**	
Host Control Register	(HCR)	B-137
Host Status Register	(HSR)	B-138
Transmit Data Register	(HTX)	B-139
Receive Data Receive	(HRX)	B-140
Interface Control Register	(ICR)	B-141
Command Vector Register	(CVR)	B-142
Interface Status Register	(ISR)	B-143
Interrupt Vector Register	(IVR)	B-144
Receive Byte Registers	(RXH, RXL)	B-145
Transmit Byte Registers	(TXH, TXL)	B-146

Application: _____ Date: _____

Programmer: _____

SIM

System Integration Module Control Register (SCR)

Bits	Name	Description	
14 - 12	BOOT MODE	This bit field is set to the value on the input pins, MODC, MODB, and MODA when the last active reset source (accept COP reset) deasserts. Its value determines boot mode executed upon reset.	
		Boot Mode 0	Bootstrap from byte-wide external memory
		Boot Mode 1	Bootstrap from SPI
		Boot Mode 2	Normal expanded mode
		Boot Mode 3	Development expanded mode
		Boot Mode 4	Bootstrap from Host Port-Single Strobe Clocking
		Boot Mode 5	Bookstrap from Host Port-Dual Strobe Clocking
		Boot Mode 6	Bootstrap from SCI
		Boot Mode 7	Reserved
6	EOnCE EBL	**Enhanced OnCE Enable**	
		0	OnCE clock to core is enabled only when the core TAP is enabled
		1	OnCE clock to core is always enabled
5	CLKOUT DBL	**Clock Out Disable**	
		0	CLKOUT output presents CLKMSTR/8 (this is half the peripheral bus clock frequency)
		1	CLKOUT output pin presents static 0
4	PRAM DBL	**Program RAM Disable**	
		0	Internal program RAM enabled
		1	Internal program RAM disabled and accesses redirected to external memory
3	DRAM DBL	**Data RAM Disable**	
		0	Internal data RAM enabled
		1	Internal data RAM disabled and accesses redirected to external memory
2	SW RST	**Software Reset**	
		To reset part, write a 1 to this bit	
1	STOP DBL	**Stop Disable**	
		0	The Stop mode will be entered when the core executes a Stop instruction
		1	The core Stop instruction will not cause entry into the Stop mode
0	WAIT DBL	**Wait Disable**	
		0	The Wait mode will be entered when the core executes a Wait instruction
		1	The core Wait instruction will not cause entry into the Wait mode

SIM Control Register (SCR) $1FFF08 +$0	Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Read	0	BOOT MODE			CHIP REV				0	EOnCE EBL	CLKOUT DBL	PRAM DBL	DRAM DBL	SW RST	STOP DBL	WAIT DBL
	Write																
	Reset	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0

denotes Reserved Bits

SIM

System Integration Module Control Data Registers 1-2 (SCD1-2)

Bits	Name	Description
15 - 0	SCD1	**Software Control Data One**
		This register is reset only by the POR and is intended for use by software developers to place data to be unaffected by other reset sources.
15 - 0	SCD2	**Software Control Data Two**
		This register is reset only by the POR and is intended for use by software developers to place data to be unaffected by other reset sources.

SIM Control Data Register 1 (SCD1) $1FFF08 + $1	Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Read	SOFTWARE CONTROL DATA 1															
	Write																
	POR	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Reset	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

SIM Control Data Register 2 (SCD2) $1FFF08 +$2	Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Read	SOFTWARE CONTROL DATA 2															
	Write																
	POR	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Reset	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

Application: _____ Date: _____

_____ Programmer: _____

EMI

Chip Select Register Base Address and Block Size (CSBAR)

Bits	Name	Description
15	ADDR23	Determines the memory map start address where the chip select is active.
14	ADDR22	
13	ADDR21	
12	ADDR20	
11	ADDR19	
10	ADDR18	
9	ADDR17	
8	ADDR16	
7	ADDR15	
6	ADDR14	
5	ADDR13	
4	ADDR12	
3 - 0	BLKSZ	Determines which bits in the base address field are compared to corresponding bits on the address bus during an access.

$\overline{CS}$ Register Base Address/Block Size (CSBAR0-CSBAR3) $1FFE40 + $0-$3	Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Read	ADDR 23	ADDR 22	ADDR 21	ADDR 20	ADDR 19	ADDR 18	ADDR 17	ADDR 16	ADDR 15	ADDR 14	ADDR 13	ADDR 12	BLKSZ			
	Write																
	Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

EMI

Chip Select Option Register (CSOR0-3)

Bits	Name	Description			
9 - 8	BYTE_EN	**Upper/Lower Byte Enable (UBS and LBS)**			
		Accesses to external data memory are typically through the use of a word. For data memory access, the 56800 core can also access bytes, yielding the upper and lower half of a word.			
		CSOR Encoding of CS $\overline{\text{UBS}}$ Functionality		CSOR Encoding of CS $\overline{\text{LBS}}$ Functionality	
		00	Disabled	00	Disabled
		01	Lower Byte Enabled	01	Lower Byte Enabled
		10	Upper Byte Enabled	10	Upper Byte Enabled
		11	Both Bytes Enabled	11	Both Bytes are Enabled
7 - 6	R/W	**Read/Write**			
		00	The chip select will be disabled		
		01	The chip select will be enabled for both read/write		
		10	The chip select will allow read only		
		11	The chip select will allow read/write		
5 - 4	PS/DS	**Program/Data Space Select**			
		00	The chip select will be disabled		
		01	The chip select will allow Data Space only		
		10	The chip select will allow Program Space only		
		11	The chip select will be enabled		
3 - 0	WWS	**Write Wait State**			
		Specifies minimum number of IPBus_CLK Wait states required by an EMI access.			

Chip Select Option Register (CSOR0-CSOR3) $1FFE40 + $8-$B	Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Read / Write	RWS					BYTE_EN		R/W		PS/DS		WWS				
	Reset	1	0	1	1	1	0	0	0	0	0	0	1	0	1	1	1

EMI

Chip Select Timing Control Registers (CSTC0–3)

Bits	Name	Description
15 - 14	WWSS	Write Wait States Setup Delay
		This field affects the write cycle timing diagram, illustrated in Figure 5-16. Additional time (clock cycles) is provided between the assertion of CS*n* and address lines and the assertion of WR. The value of WWSS should be set as indicated in Section 5.7.2.
13 - 12	WWSH	Write Wait States Hold Delay
		This field affects the write cycle timing diagram, illustrated in Figure 5-17. The WWSH field specifies the number of additional system clocks to hold the address, data, and CS*n* signals after the WR signal is deasserted. The value of WWSH should be set as indicated in Section 5.7.2.
11 - 10	RWSS	Read Wait States Setup Delay
		This field affects the read cycle timing diagram, illustrated in Figure 5-10. Additional time (clock cycles) is provided between the assertion of CS*n* and address lines and the assertion of RD. The value of RWSS should be set as indicated in Section 5.7.1.
9 - 8	RWSH	Read Wait States Hold Delay
		This field affects the read cycle timing diagram, illustrated in Figure 5-11. The RWSH field specifies the number of additional system clocks to hold the address, data, and CS*n* signals after the RD signal is deasserted. The value of RWSH should be set as indicated in Section 5.7.1.
2 - 0	MDAR	Minimal Delay After Read
		This field specifies the number of system clocks to delay between reading from memory in a CS*n* controlled space and reading from another device. Since a write to the device implies activating the device on the bus, this is also considered a read from another device. Figure 5-6 illustrates the timing issue requiring the introduction of the MDAR field. In this diagram, CS1 is assumed to operate a slow flash memory in P-space while CS2 is operating a faster RAM in X-space. In some bus contention cases, it is possible to encounter data integrity problems where the contention is occurring at the time the data bus is sampled.

Chip Select Timing Control Register (CSTC0-CSOR3) $1FFE40 + $10-$13	Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Read	WWSS		WWSH		RWSS		RWSH		0	0	0	0	0	MDAR		
	Write																
	Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

denotes Reserved Bits

Application: _____ Date: _____

_____ Programmer: _____

EMI

Bus Control Register (BCR)

Bits	Name	Description
15	DRV	**Drive**
		This control bit is used to specify what occurs on the external memory port pins when no external access is performed. For example, it determines whether pins are placed in tri-state or remain driven.
14 - 12	BMDAR	**Base Minimal Delay After Read**
		This bit field specifies the number of system clocks to delay after reading from memory not in $\overline{CS}$ controlled space. Since a write to the device implies activating the device on the bus, this is also considered a read from another device, therefore activating the BMDAR timing control. Please see the description of the MDAR field of the CSTC registers for a discussion of the function of this control.
9-5	BWWS	**Base Write Wait States**
		This bit field specifies the number of additional system clocks 0-30 (31 is invalid) to delay for write access to the selected memory when the memory address does *not* fall within $\overline{CS}$ controlled range. The value of BWWS should be set as indicated in **Section 5.7.**
4-0	BRWS	**Base Read Wait States**
		This bit field specifies the number of additional system clocks 0-30 (31 is invalid) to delay for read access to the selected memory when the memory address does *not* fall within $\overline{CS}$ controlled range. The value of BRWS should be set as indicated in **Section 5.7.**

Bus Control Register (BCR) $1FFE40 +$18	Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Read	DRV	BMDAR			0	0	BWWS					BRWS				
	Write																
	Reset	0	0	0	0	0	0	1	0	1	1	1	1	0	1	1	1

denotes Reserved Bits

Application: _____ Date: _____

_____ Programmer: _____

OCCS

CGM Control Register (CGMCR)

Bits	Name	Description
13	LCK1	**Lock 1 Status**
		This bit shows the status of the lock detector state for the LCK1 circuit.
		0 PLL not locked
		1 PLL locked
12	LCK0	**Lock 0 Status**
		This bit shows the status of the lock detector state for the LCK0 circuit.
		0 PLL not locked
		1 PLL locked
11	SEL	**Clock Source Select**
		This bit is used to control the source of the master clock to the SIM.
		0 Oscillator output selected (default)
		1 PLL output selected
6 - 5	LCK1_IE	**Lock 1 Interrupt Enable**
		This is an optional interrupt bit.
		00 Disable interrupt (default)
		01 Enable interrupt on rising edge of LCK1
		10 Enable interrupt on falling edge of LCK1
		11 Enable interrupt on any edge of LCK1

CGM Control Register (CGMCR) $1FFF10 + $0	Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Read	0	0	LCK1	LCK0	SEL	0	0	0	0	LCK1_IE		LCK0_IE		LCKON	TOD_SEL	PDN
	Write																
	Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

denotes Reserved Bits

See the following page for continuation of this register

OCCS

CGM Control Register (CGMCR) continued

Bits	Name	Description	
4 - 3	**LCK0_IE**	**Lock 0 Interrupt Enable**	
		This is an optional interrupt bit.	
		00	Disable interrupt (default)
		01	Enable interrupt on rising edge of LCK0
		10	Enable interrupt on falling edge of LCK0
		11	Enable interrupt on any edge of LCK0
2	**LCKON**	**Lock Detector On**	
		This is an optional interrupt bit.	
		0	Lock detector disabled (default)
		1	Lock detector enabled
1	**TOD_SEL**	**Time of Day Select**	
		This bit is used to select between the two possible TOD_SEL sources.	
		0	TOD_CLK is generated by the oscillator (default)
		1	TOD_CLK is generated by the CGM
0	**PDN**	**The PLL Power-Down**	
		This bit can be turned off by setting the power-down bit to 1.	
		0	PLL turned on
		1	PLL powered down (default)

CGM Control Register (CGMCR) $1FFF10 + $0	Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Read	0	0	LCK1	LCK0	SEL	0	0	0	0	LCK1_IE		LCK0_IE		LCKON	TOD_SEL	PDN
	Write																
	Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

denotes Reserved Bits

OCCS

CGM Divide-By Register (CGMCR)

Bits	Name	Description	
15 - 13	POST	**PLL Post Scaler**	
		The output of the PLL is postscaled by 1-128 based on this field. To change this field, set the SEL bit to choose the oscillator output, then this field is changed. The SEL bit is then returned to selecting the PLL postscaled output.	
		000	PLL output is divided by 1 (default)
		001	PLL output is divided by 2
		010	PLL output is divided by 4
		011	PLL output is divided by 8
		100	PLL output is divided by 16
		101	PLL output is divided by 32
		110	PLL output is divided by 64
		111	PLL output is divided by 128
6 - 0	PLLDB	**PLL Divide-By**	
		The PLL output frequency is controlled by the PLL divide-by value. Each time a new value is written into the PLLDB field, the Lock Detector circuit is reset. Before changing the divide-by, set the SEL bit to choose the oscillator output.	

CGM Divide-By Register (CGMDB) $1FFF10 + $1	Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Read	POST			0	0	0	0	0	0	PLLDB						
	Write																
	Reset	0	0	0	0	0	0	0	0	0	0	1	1	1	0	1	1

denotes Reserved Bits

OCCS

CGM Time-of-Day Register (CGMTOD)

Bits	Name	Description
11 - 0	TOD	**Time-of-Day**
		The output of the oscillator is divided by (TOD + 1) and then divided by 2 to generate the TOD clock used by the COP module when TOD_SEL is high. The value of TOD should be chosen to result in a TOD clock frequency in the range of 15.12KHz to 31.25KHz. This register is only reset during Power-On Reset (POR).

CGM Time of Day Register (CGMTOD) $1FFFF10 + $2	Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Read	0	0	0	0	TOD											
	Write																
	Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

denotes Reserved Bits

Application: _____ Date: _____

Programmer: _____

COP

COP Control Register (COPCTL)

Bits	Name	Description
4	BYPS	**BYPASS (For factory use only)**
		When this bit is set, it allows factory testing of the COP is accelerated by routing the IPBus clock to the counter instead of the OSCCLK. This bit should not be set during normal chip operation.
3	CSEN	**COP Stop Enable**
		This bit controls the operation of the COPcounter Stop mode. It can be changed only when the CWP bit is set to zero.
2	CWEN	**COP Wait Enable**
		This bit controls the operation of the COP counter in the Wait mode. It can be changed only when the CWP bit is set to zero.
1	CEN	**COP Enable**
		This bit controls the operation of the COP counter. This bit can only be changed when CWP is set to zero. This bit always reads as zero when the chip is in the Debug mode.
0	CWP	**COP Write Protect**
		This bit controls the write protection feature of the COP Control (COPCTL) and the COP Timeout (COPTO) registers. Once set, this bit can only be cleared by resetting the module.

COP Control Register (COPCTL) $1FFFD0 + $0	Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Read	0	0	0	0	0	0	0	0	0	0	0	BYPS	CSEN	CWEN	CEN	CWP
	Write																
	Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

denotes Reserved Bits

COP

COP Time-Out Register (COPTO)

Bits	Name	Description
15 - 0	TIMEOUT	**COP Time-Out Period**
		This register determines the time-out period of the COP counter. TIMEOUT should be written before the COP is enabled. Once the COP is enabled, the recommended procedure for changing TIMEOUT is to disable the COP, write to COPTO, then re-enable the COP, ensuring the new TIMEOUT is loaded into the counter. Alternatively, the CPU can write to COPTO, then write the proper patterns to COPCTR, causing the counter to reload with the new TIMEOUT value. The COP counter is not reset by a write to COPTO. Changing TIMEOUT while the COP is enabled will result in a time-out period differing from the expected value. These bits can only be changed when the CWP bit is set to zero.

COP Timeout Register (COPTO) $1FFFD0 + $1	Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Read	TIMEOUT															
	Write																
	Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

COP

COP Counter Register (COPCTR)

Bits	Name	Description
15 - 0	COPCTR	**COP Counter(Count)**
		This is the current value of the COP counter as it counts down from the timeout value to zero. A reset is issued when this count reaches zero.
15 - 0	COPCTR	**COP Counter (Service)**
		When enabled, the COP requires a service sequence be performed periodically in order to clear the COP counter and prevent a reset from being issued. This routine consists of writing $5555 to the COPCTR followed by writing $AAAA before the timeout period expires. The writes to COPCTR must be performed in the correct order, but any number of other instructions, and writes to other registers, may be executed between the two writes.

COP Counter Register (COPCTR) $1FFFD0 + $2	Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Read	COUNT															
	Write	SERVICE															
	Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Application: _____ Date: _____

_____ Programmer: _____

ITCN

Interrupt Priority Register 0 (IPR0)

Bits	Name	Description	
13 - 12	BKPT_U0 IPL	**Breakpoint Unit 0 EOnCE Interrupt Priority Level**	
		This bit field is used to set the interrupt priority levels for this EOnCE IRQ.	
		00	IRQ disabled by default
		01	IRQ is priority level 1
		10	IRQ is priority level 2
		11	IRQ is priority level 3
11 - 10	STPCENT IPL	**EOnCE Step Counter Interrupt Priority Level**	
		This bit field is used to set the interrupt priority levels for this EOnCE IRQ.	
		00	IRQ disabled by default
		01	IRQ is priority level 1
		10	IRQ is priority level 2
		11	IRQ is priority level 3

Interrupt Priority Register 0 (IPR0) $1FFF20 + $0	Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Read	0	0	BKPT_U0 IPL		STPCNT IPL		0	0	0	0	0	0	0	0	0	0
	Write																
	Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

denotes Reserved Bits

ITCN

Interrupt Priority Register 1 (IPR1)

Bits	Name	Description	
5 - 4	RX_REG IPL	**Receive Data Empty Register Interrupt Priority Level**	
		This bit field is used to set the interrupt priority levels for this OnCE IRQ.	
		00	IRQ disabled by default
		01	IRQ is priority level 1
		10	IRQ is priority level 2
		11	IRQ is priority level 3
3 - 2	TX_REG IPL	**Transmit Data Full Interrupt Priority Level**	
		This bit field is used to set the interrupt priority levels for this OnCE IRQ.	
		00	IRQ disabled by default
		01	IRQ is priority level 1
		10	IRQ is priority level 2
		11	IRQ is priority level 3
1 - 0	TRBUF IPL	**Trace Buffer Interrupt Priority Level**	
		This bit field is used to set the interrupt priority levels for this OnCE IRQ.	
		00	IRQ disabled by default
		01	IRQ is priority level 1
		10	IRQ is priority level 2
		11	IRQ is priority level 3

Interrupt Priority Register 1 (IPR1) $1FFF20 + $1	Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Read	0	0	0	0	0	0	0	0	0	0	RX_REG IPL		TX_REG IPL		TRBUF IPL	
	Write																
	Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

denotes Reserved Bits

ITCN

Interrupt Priority Register 2 (IPR2)

Bits	Name	Description	
15 - 14	DMA2 IPL	**Done Interrupt Priority Level 2**	
		These bit fields are used to set the interrupt priority levels for certain EOnCE IRQs.	
		00	IRQ disabled by default
		01	IRQ is priority level 1
		10	IRQ is priority level 2
		11	IRQ is priority level 3
13 -12	DMA1 IPL	**Done Interrupt Priority Level 1**	
		These bit fields are used to set the interrupt priority levels for certain EOnCE IRQs.	
		00	IRQ disabled by default
		01	IRQ is priority level 1
		10	IRQ is priority level 2
		11	IRQ is priority level 3
11 -10	DMA0 IPL	**Done Interrupt Priority Level 0**	
		These bit fields are used to set the interrupt priority levels for certain EOnCE IRQs.	
		00	IRQ disabled by default
		01	IRQ is priority level 1
		10	IRQ is priority level 2
		11	IRQ is priority level 3

Interrupt Priority Register 2 (IPR2) $1FFF20 + $2	Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Read	DMA2 IPL		DMA1 IPL		DMA0 IPL		0	0	LOCK IPL		0	0	IRQB IPL		IRQA IPL	
	Write																
	Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

denotes Reserved Bits

See the following page for continuation of this register

ITCN

Interrupt Priority Register 2 (IPR2) continued

Bits	Name	Description
7 - 6	**LOCK IPL**	**Loss of Lock Interrupt Priority Level**
		This bit field is used to set the interrupt priority levels for this peripheral IRQ.
		00 IRQ disabled by default
		01 IRQ is priority level 1
		10 IRQ is priority level 2
		11 IRQ is priority level 3
3 - 2	**IRQB IPL**	**External IRQB Interrupt Priority Level**
		This bit field is used to set the interrupt priority levels for this peripheral IRQ.
		00 IRQ disabled by default
		01 IRQ is priority level 1
		10 IRQ is priority level 2
		11 IRQ is priority level 3
1 - 0	**IRQA IPL**	**External IRQA Interrupt Priority Level**
		This bit field is used to set the interrupt priority levels for this peripheral IRQ.
		00 IRQ disabled by default
		01 IRQ is priority level 1
		10 IRQ is priority level 2
		11 IRQ is priority level 3

Interrupt Priority Register 2 (IPR2) $1FFF20 + $2	Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Read	DMA2 IPL		DMA1 IPL		DMA0 IPL		0	0	LOCK IPL		0	0	IRQB IPL		IRQA IPL	
	Write																
	Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

denotes Reserved Bits

ITCN

Interrupt Priority Register 3 (IPR3)

Bits	Name	Description
15 - 14	ESSI0_TD IPL	**ESSI0 Transmit Data Interrupt Priority Level**
		This bit field is used to set the interrupt priority levels for this peripheral IRQ.
		00 IRQ disabled by default
		01 IRQ is priority level 0
		10 IRQ is priority level 1
		11 IRQ is priority level 2
13 - 12	ESSI0_TDES IPL	**ESSI0 Transmit Data with Exception Status Interrupt Priority Level**
		This bit field is used to set the interrupt priority levels for this peripheral IRQ.
		00 IRQ disabled by default
		01 IRQ is priority level 0
		10 IRQ is priority level 1
		11 IRQ is priority level 2
11 - 10	ESSIO_RLS IPL	**ESSIO Receive Last Slot Interrupt Priority Level**
		This bit field is used to set the interrupt priority levels for certain peripheral IRQs.
		00 IRQ disabled by default
		01 IRQ is priority level 0
		10 IRQ is priority level 1
		11 IRQ is priority level 2
9 - 8	ESSI0_RD IPL	**ESSI0 Receive Data Interrupt Priority Level**
		This bit field is used to set the interrupt priority levels for this peripheral IRQ.
		00 IRQ disabled by default
		01 IRQ is priority level 0
		10 IRQ is priority level 1
		11 IRQ is priority level 2

Interrupt Priority Register 3 (IPR3) $1FFF20 + $3	Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Read / Write	ESSI0_TD IPL		ESSI0_TDES IPL		ESSIO_RLS IPL		ESSI0_RLS IPL		ESSI_RDES IPL		DMA5 IPL		DMA4 IPL		DMA3 IPL	
	Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

denotes Reserved Bits

See the following page for continuation of this register

ITCN

Interrupt Priority Register 3 (IPR3) continued

Bits	Name	Description	
7 - 6	ESSI0_RDES IPL	**ESSI0 Receive Data with Exception Status Interrupt Priority Level**	
		This bit field is used to set the interrupt priority levels for this peripheral IRQ.	
		00	IRQ disabled by default
		01	IRQ is priority level 0
		10	IRQ is priority level 1
		11	IRQ is priority level 2
5 - 4	DMA5 IPL	**Done Interrupt Priority Level**	
		This bit field is used to set the interrupt priority levels for certain peripheral IRQs.	
		00	IRQ disabled by default
		01	IRQ is priority level 0
		10	IRQ is priority level 1
		11	IRQ is priority level 2
3 - 2	DMA4 IPL	**Done Interrupt Priority Level**	
		This bit field is used to set the interrupt priority levels for certain peripheral IRQs.	
		00	IRQ disabled by default
		01	IRQ is priority level 0
		10	IRQ is priority level 1
		11	IRQ is priority level 2
1 - 0	DMA3 IPL	**Done Interrupt Priority Level**	
		This bit field is used to set the interrupt priority levels for certain peripheral IRQs.	
		00	IRQ disabled by default
		01	IRQ is priority level 0
		10	IRQ is priority level 1
		11	IRQ is priority level 2

Interrupt Priority Register 3 (IPR3) $1FFF20 + $3	Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Read	ESSI0_TD IPL		ESSI0_TDES IPL		ESSIO_RLS IPL		ESSI0_RLS IPL		ESSI_RDES IPL		DMA5 IPL		DMA4 IPL		DMA3 IPL	
	Write																
	Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ITCN

Interrupt Priority Register 4 (IPR4)

Bits	Name	Description	
15 - 14	SPI_RCV IPL	**SPI Receiver Full Interrupt Priority Level**	
		This bit field is used to set the interrupt priority levels for this peripheral IRQs.	
		00	IRQ disabled by default
		01	IRQ is priority level 0
		10	IRQ is priority level 1
		11	IRQ is priority level 2
13 -12	ESSI1_TLS IPL	**ESSI1 Transmit Last Slot Interrupt Priority Level**	
		This bit field is used to set the interrupt priority levels for certain peripheral IRQs.	
		00	IRQ disabled by default
		01	IRQ is priority level 0
		10	IRQ is priority level 1
		11	IRQ is priority level 2
11 - 10	ESSI1_TD IPL	**ESSI1 Transmit Data Interrupt Priority Level**	
		This bit field is used to set the interrupt priority levels for certain peripheral IRQs.	
		00	IRQ disabled by default
		01	IRQ is priority level 0
		10	IRQ is priority level 1
		11	IRQ is priority level 2
9 - 8	ESSI1_TDES IPL	**ESSI1Receive Data with Exception Status Interrupt Priority Level**	
		This bit field is used to set the interrupt priority levels for certain peripheral IRQs.	
		00	IRQ disabled by default
		01	IRQ is priority level 0
		10	IRQ is priority level 1
		11	IRQ is priority level 2

Interrupt Priority Register 4 (IPR4) $1FFF20 + $4	Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Read	SPI_RCV IPL		ESSI1_TLS IPL		ESSI1_TD IPL		ESSI1_TD ES IPL		ESSI1_RLS IPL		ESSI1_RD IPL		ESSI1_RD ES IPL		ESSI0_TLS IPL	
	Write																
	Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

denotes Reserved Bits

See the following page for continuation of this register

ITCN

Interrupt Priority Register 4 (IPR4) continued

Bits	Name	Description	
7 - 6	ESSI1_RLS IPL	**SPI Receive Last Slot Interrupt Priority Level**	
		This bit field is used to set the interrupt priority levels for this peripheral IRQs.	
		00	IRQ disabled by default
		01	IRQ is priority level 0
		10	IRQ is priority level 1
		11	IRQ is priority level 2
5 -4	ESSI1_RD IPL	**ESSI1 Receive Data Interrupt Priority Level**	
		This bit field is used to set the interrupt priority levels for certain peripheral IRQs.	
		00	IRQ disabled by default
		01	IRQ is priority level 0
		10	IRQ is priority level 1
		11	IRQ is priority level 2
3 - 2	ESSI1_RDES IPL	**ESSI1 Receive Data with Exception Status Interrupt Priority Level**	
		This bit field is used to set the interrupt priority levels for certain peripheral IRQs.	
		00	IRQ disabled by default
		01	IRQ is priority level 0
		10	IRQ is priority level 1
		11	IRQ is priority level 2
1 - 0	ESSI0_TLS IPL	**ESSI0 Transmit last Slot Interrupt Priority Level**	
		This bit field is used to set the interrupt priority levels for certain peripheral IRQs.	
		00	IRQ disabled by default
		01	IRQ is priority level 0
		10	IRQ is priority level 1
		11	IRQ is priority level 2

Interrupt Priority Register 4 (IPR4) $1FFF20 + $4	Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Read	SPI_RCV IPL		ESSI1_TLS IPL		ESSI1_TD IPL		ESSI1_TD ES IPL		ESSI1_RLS IPL		ESSI1_RD IPL		ESSI1_RD ES IPL		ESSI0_TLS IPL	
	Write																
	Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

denotes Reserved Bits

ITCN

Interrupt Priority Register 5 (IPR5)

Bits	Name	Description
15 - 14	HOST_XMIT IPL	**SCI Receive Full Interrupt Priority Level**
		This bit field is used to set the interrupt priority levels for this peripheral IRQ.
		00 IRQ disabled by default
		01 IRQ is priority level 0
		10 IRQ is priority level 1
		11 IRQ is priority level 2
13 - 12	HOST_RCV IPL	**SCI Receive Error Interrupt Priority Level**
		This bit field is used to set the interrupt priority levels for this peripheral IRQ.
		00 IRQ disabled by default
		01 IRQ is priority level 0
		10 IRQ is priority level 1
		11 IRQ is priority level 2
11 - 10	SCI0_RCV IPL	**SCI Receive Idle Interrupt Priority Level**
		This bit field is used to set the interrupt priority levels for this peripheral IRQ.
		00 IRQ disabled by default
		01 IRQ is priority level 0
		10 IRQ is priority level 1
		11 IRQ is priority level 2
9 - 8	SCI0_RERR IPL	**SCI0 Transmitter Idle Interrupt Priority Level**
		This bit field is used to set the interrupt priority levels for certain peripheral IRQs.
		00 IRQ disabled by default
		01 IRQ is priority level 0
		10 IRQ is priority level 1
		11 IRQ is priority level 2

Interrupt Priority Register 5 (IPR5) $1FFF20 + $5	Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Read / Write	HOST_XMIT IPL		HOST_RCV IPL		SCI0_RCV IPL		SCI0_RERR IPL		SC0_RIDL IPL		SCI0_TDL IPL		SCI0_XMIT IPL		SPI_XMIT IPL	
	Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

See the following page for continuation of this register

ITCN

Interrupt Priority Register 5 (IPR5) continued

Bits	Name	Description	
7 - 6	SCI0_RIDL IPL	**SCI0 Receiver Idle Interrupt Priority Level**	
		This bit field is used to set the interrupt priority levels for this peripheral IRQ.	
		00	IRQ disabled by default
		01	IRQ is priority level 0
		10	IRQ is priority level 1
		11	IRQ is priority level 2
5 - 4	SCI0_TIDL IPL	**SCI Transmitter Idle Interrupt Priority Level**	
		This bit field is used to set the interrupt priority levels for this peripheral IRQ.	
		00	IRQ disabled by default
		01	IRQ is priority level 0
		10	IRQ is priority level 1
		11	IRQ is priority level 2
3 - 2	SCI0_XMIT IPL	**SPI Transmitter Empty Interrupt Priority Level**	
		This bit field is used to set the interrupt priority levels for this peripheral IRQ.	
		00	IRQ disabled by default
		01	IRQ is priority level 0
		10	IRQ is priority level 1
		11	IRQ is priority level 2
1 - 0	SPI_XMIT IPL	**SPI Transmitter Empty Interrupt Priority Level**	
		This bit field is used to set the interrupt priority levels for this peripheral IRQ.	
		00	IRQ disabled by default
		01	IRQ is priority level 0
		10	IRQ is priority level 1
		11	IRQ is priority level 2

Interrupt Priority Register 5 (IPR5) $1FFF20 + $5	Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Read	HOST_XMIT IPL		HOST_RCV IPL		SCI0_RCV IPL		SCI0_RERR IPL		SC0_RIDL IPL		SCI0_TIDL IPL		SCI0_XMIT IPL		SPI_XMIT IPL	
	Write																
	Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ITCN

Interrupt Priority Register 6 (IPR6)

Bits	Name	Description	
15 - 14	TOVF1 IPL	**Timer Overflow 1 Interrupt Priority Level**	
		These two bits are used to set the interrupt priority levels for this peripheral IRQ.	
		00	IRQ disabled by default
		01	IRQ is priority level 0
		10	IRQ is priority level 1
		11	IRQ is priority level 2
13 -12	TCMP1 IPL	**Timer Compare 1 Interrupt Priority Level**	
		These two bits are used to set the interrupt priority levels for this peripheral IRQ.	
		00	IRQ disabled by default
		01	IRQ is priority level 0
		10	IRQ is priority level 1
		11	IRQ is priority level 2
11 - 10	TINP0 IPL	**Timer Input Edge 0 Interrupt Priority Level**	
		These two bits are used to set the interrupt priority levels for this peripheral IRQ.	
		00	IRQ disabled by default
		01	IRQ is priority level 0
		10	IRQ is priority level 1
		11	IRQ is priority level 2
9 - 8	TOVF0 IPL	**Timer Overflow 0 Interrupt Priority Level**	
		This bit field is used to set the interrupt priority levels for this peripheral IRQ.	
		00	IRQ disabled by default
		01	IRQ is priority level 0
		10	IRQ is priority level 1
		11	IRQ is priority level 2

Interrupt Priority Register 6 (IPR6) $1FFF20 +$6	Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Read	TOVF1 IPL		TCMP1 IPL		TINP0 IPL		TOVF0 IPL		TCMP0 IPL		TOD IPL		TOD_ALRM IPL		TOD_CMD IPL	
	Write																
	Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

See the following page for continuation of this register

ITCN

Interrupt Priority Register 6 (IPR6) continued

Bits	Name	Description	
7 -6	TCMP0 IPL	**Timer Compare 0 Interrupt Priority Level**	
		This bit field is used to set the interrupt priority levels for this peripheral IRQ.	
		00	IRQ disabled by default
		01	IRQ is priority level 0
		10	IRQ is priority level 1
		11	IRQ is priority level 2
5 - 4	TOD IPL	**Timer Compare 0 Interrupt Priority Level**	
		This bit field is used to set the interrupt priority levels for this peripheral IRQ.	
		00	IRQ disabled by default
		01	IRQ is priority level 0
		10	IRQ is priority level 1
		11	IRQ is priority level 2
3 - 2	TOD_ALRM IPL	**Time of Day Alarm Interrupt Priority Level**	
		This bit field is used to set the interrupt priority levels for this peripheral IRQ.	
		00	IRQ disabled by default
		01	IRQ is priority level 0
		10	IRQ is priority level 1
		11	IRQ is priority level 2
1 - 0	HOST_CMD IPL	**Host I/F Command Interrupt Priority Level**	
		This bit field is used to set the interrupt priority levels for this peripheral IRQ.	
		00	IRQ disabled by default
		01	IRQ is priority level 0
		10	IRQ is priority level 1
		11	IRQ is priority level 2

Interrupt Priority Register 6 (IPR6) $1FFF20 + $6	Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Read	TOVF1 IPL		TCMP1 IPL		TINP0 IPL		TOVF0 IPL		TCMP0 IPL		TOD IPL		TOD_ALRM IPL		HOST_CMD IPL	
	Write																
	Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ITCN

Interrupt Priority Register 7 (IPR7)

Bits	Name	Description
13 - 12	TINP3 IPL	**Timer Input Edge 3 Interrupt Priority Level**
		This bit field is used to set the interrupt priority levels for this peripheral IRQ.
		00 IRQ disabled by default
		01 IRQ is priority level 0
		10 IRQ is priority level 1
		11 IRQ is priority level 2
11 - 10	TOVF3 IPL	**Timer Overflow 3 Interrupt Priority Level**
		This bit field is used to set the interrupt priority levels for this peripheral IRQ.
		00 IRQ disabled by default
		01 IRQ is priority level 0
		10 IRQ is priority level 1
		11 IRQ is priority level 2
9 - 8	TCMP3 IPL	**Timer Compare 3 Interrupt Priority Level**
		This bit field is used to set the interrupt priority levels for this peripheral IRQ.
		00 IRQ disabled by default
		01 IRQ is priority level 0
		10 IRQ is priority level 1
		11 IRQ is priority level 2
7 - 6	TINP2 IPL	**Timer Input Edge 2 Interrupt Priority Level**
		This bit field is used to set the interrupt priority levels for this peripheral IRQ.
		00 IRQ disabled by default
		01 IRQ is priority level 0
		10 IRQ is priority level 1
		11 IRQ is priority level 2

Interrupt Priority Register 7 (IPR7) $1FFF20 + $7	Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Read	0	0	TINP3 IPL		TOVF3 IPL		TCMP3 IPL		TINP2 IPL		TOVF2 IPL		TCMP2 IPL		TINP1 IPL	
	Write																
	Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

denotes Reserved Bits

See the following page for continuation of this register

ITCN

Interrupt Priority Register 7 (IPR7) continued

Bits	Name	Description	
5 - 4	TOVF2 IPL	**Timer Overflow 2 Interrupt Priority Level**	
		This bit field is used to set the interrupt priority levels for this peripheral IRQ.	
		00	IRQ disabled by default
		01	IRQ is priority level 0
		10	IRQ is priority level 1
		11	IRQ is priority level 2
3 - 2	TCMP2 IPL	**Timer Compare 2 Interrupt Priority Level**	
		This bit field is used to set the interrupt priority levels for this peripheral IRQ.	
		00	IRQ disabled by default
		01	IRQ is priority level 0
		10	IRQ is priority level 1
		11	IRQ is priority level 2
1 - 0	TINP1 IPL	**Timer Input Edge 1 Interrupt Priority Level**	
		This bit field is used to set the interrupt priority levels for this peripheral IRQ.	
		00	IRQ disabled by default
		01	IRQ is priority level 0
		10	IRQ is priority level 1
		11	IRQ is priority level 2

Interrupt Priority Register 7 (IPR7) $1FFF20 + $7	Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Read	0	0	TINP3 IPL		TOVF3 IPL		TCMP3 IPL		TINP2 IPL		TOVF2 IPL		TCMP2 IPL		TINP1 IPL	
	Write																
	Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

denotes Reserved Bits

ITCN

Interrupt Priority Register 8 (IPR8)

Bits	Name	Description	
9 - 8	SCI1_RCV IPL	**SCI1 Receiver Full interrupt Priority Level**	
		This bit field is used to set the interrupt priority levels for this peripheral IRQ. These IRQs are limited to priorities 0-2 and are disabled by default.	
		00	IRQ disabled by default
		01	IRQ is priority level 0
		10	IRQ is priority level 1
		11	IRQ is priority level 2
7 - 6	SCI1_RERR IPL	**SCI1 Receiver Error Interrupt Priority Level**	
		This bit field is used to set the interrupt priority levels for this peripheral IRQ.These IRQs are limited to priorities 0-2 and are disabled by default.	
		00	IRQ disabled by default
		01	IRQ is priority level 0
		10	IRQ is priority level 1
		11	IRQ is priority level 2
5 - 4	SCI1_RIDL IPL	**SCI1 Receiver Idle Interrupt Priority Level**	
		This bit field is used to set the interrupt priority levels for this peripheral IRQ. These IRQs are limited to priorities 0-2 and are disabled by default.	
		00	IRQ disabled by default
		01	IRQ is priority level 0
		10	IRQ is priority level 1
		11	IRQ is priority level 2

Interrupt Priority Register 8 (IPR8) $1FFF20 + $8	Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Read	0	0	0	0	0	0	SCI1_RCV IPL		SCI1_RERR IPL		SCI1_RIDL IPL		SCI1_TIDL IPL		SCI1_XMIT IPL	
	Write																
	Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

denotes Reserved Bits

See the following page for continuation of this register

ITCN

Interrupt Priority Register 8 (IPR8) continued

Bits	Name	Description	
3 - 2	SCI1_TIDL IPL	**SCI1 Receiver Idle Interrupt Level**	
		This bit field is used to set the interrupt priority levels for this peripheral IRQ. These IRQs are limited to priorities 0-2 and are disabled by default.	
		00	IRQ disabled by default
		01	IRQ is priority level 0
		10	IRQ is priority level 1
		11	IRQ is priority level 2
1 - 0	SCI1_TIDL IPL	**SCI1 Transmitter Idle Interrupt Priority Level**	
		This bit field is used to set the interrupt priority levels for this peripheral IRQ.These IRQs are limited to priorities 0-2 and are disabled by default.	
		00	IRQ disabled by default
		01	IRQ is priority level 0
		10	IRQ is priority level 1
		11	IRQ is priority level 2

Interrupt Priority Register 8 (IPR8) $1FFF20 + $8	Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Read	0	0	0	0	0	0	SCI1_RCV IPL		SCI1_RERR IPL		SCI1_RIDL IPL		SCI1_TIDL IPL		SCI1_XMIT IPL	
	Write																
	Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

denotes Reserved Bits

ITCN

Vector Base Address Register (VBA)

Bits	Name	Description
12 - 0	VBA	**Vector Base Address**
		The value in this register is used as the upper 13 bits of the interrupt vector VAB[20:0]. The lower eight bits are determined based on the highest priority interrupt, which are appended onto VBA before presenting the full VAB to the core.

Vector Base Address Register (VBA) $1FFF20 + $9	Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Read	0	0	0	VECTOR BASE ADDRESS												
	Write																
	Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

denotes Reserved Bits

ITCN

Fast Interrupt Match Register 0 (FIM0)

Bits	Name	Description
6 - 0	FIM0	**Fast Interrupt Match 0**
		This value is used to declare which two IRQs will be Fast Interrupts. Fast Interrupt vector directly to a service routine based on values in the Fast Interrupt Vector Address registers without having to go to a jump table first. IRQs used as fast interrupts must be set to priority level two. Unexpected results will occur if a fast interrupt vector is set to any other priority. Fast interrupts automatically become the highest priority level two interrupts regardless of their actual location in the interrupt table prior to being declared fast interrupts. Fast Interrupt 0 has priority over Fast Interrupt 1.

Fast Interrupt Match Register 0 (FIM0) $1FFF20 + $A	Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Read	0	0	0	0	0	0	0	0	0	FAST INTERRUPT 0						
	Write																
	Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

denotes Reserved Bits

ITCN

Fast Interrupt Match Register 1 (FIM1)

Bits	Name	Description
6 - 0	FIM1	**Fast Interrupt Match 1**
		This value is used to declare which two IPQs will be Fast Interrupts. Fast Interrupt vector directly to a service routine based on values in the Fast Interrupt Vector Address registers without having to go to a jump table first. IRQs used as fast interrupts must be set to priority level two. Unexpected results will occur if a fast interrupt vector is set to any other priority. Fast interrupts automatically become the highest priority level two interrupts regardless of their actual location in the interrupt table prior to being declared fast interrupts. Fast Interrupt 0 has priority over Fast Interrupt 1.

Fast Interrupt Match Register 1 (FIM1) $1FFF20 + $D	Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Read	0	0	0	0	0	0	0	0	0	FAST INTERRUPT 1						
	Write																
	Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

denotes Reserved Bits

ITCN

Fast Interrupt Vector 0 Address Low and High 0 (FIVAL0, FIVAH0)

Bits	Name	Description
15 - 0	FIVAL0	**Fast Interrupt Vector Address Low 0**
		This register is combined with the FIVAH0 register to forma 21-bit vector address for the fast interrupt defined in the FIVAL0 and FIVAH0 registers. Lower 16 bits of vector address for fast interrupt 0.

Fast Interrupt Vector Address Low 0 (FIVAL0) $1FFF20 + $B	Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Read	FAST INTERRUPT 0 VECTOR ADDRESS LOW															
	Write																
	Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

BITS	NAME	Description
4 - 0	FIVAH0	**Fast Interrupt Vector Address High 0**
		This register is combined with the FIVAL0 register to form a 21-bit vector address for the fast interrupt defined in the FIVAL0 and FIVAH0 registers. Upper 5 bits of vector address for fast interrupt 0.

Fast Interrupt Vector Address High 0 (FIVAH0) $1FFF20 + $C	Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Read	0	0	0	0	0	0	0	0	0	0	0	FAST INTERRUPT 0 VECTOR ADDRESS HIGH				
	Write																
	Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

denotes Reserved Bits

ITCN

Fast Interrupt 1 Vector Address Low and High 1 (FIVAL1, FIVAH1)

Bits	Name	Description
15 - 0	FIVAL1	**Fast Interrupt 1 Vector Address Low**
		This register is combined with the FIVAH1 register to form A21-bit vector address for the fast interrupt defined in the FIVAL1 and FIVAH1 registers. Lower 16 bits of vector address for fast interrupt 1.

Fast Interrupt 1 Vector Address Low (FIVAL1) $1FFF20 + $E	Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Read	FAST INTERRUPT 1 VECTOR ADDRESS LOW															
	Write																
	Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description
4 - 0	FIVAH1	**Fast Interrupt 1 Vector Address High**
		This register is combined with the FIVAL1 register to form A 21-bit vector address for the fast interrupt defined in the FIVAL1 and FIVAH1 registers. Upper 5 bits of vector address for fast interrupt 1.

Fast Interrupt 1 Vector Address High (FIVAH1) $1FFF20 + $F	Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Read	0	0	0	0	0	0	0	0	0	0	0	FAST INTERRUPT 1 VECTOR ADDRESS HIGH				
	Write																
	Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

denotes Reserved Bits

ITCN

IRQ Pending Registers 0 - 3 (IRQP0 - IRQP3)

Bits	Name	Description	
64 - 1	IRQP0 - 3	**IRQ Pending Registers**	
		These registers combine to show the status of interrupt requests 2 through 64.	
		0	IRQ pending for this vector number
		1	No IRQ pending for this vector number

IRQ Pending Register 0 (IRQP0) $1FFF20 + $10

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	PENDING [16:1]															1
Write																
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

IRQ Pending Register 1 (IRQP1) $1FFF20 + $11

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	PENDING [32:17]															
Write																
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

IRQ Pending Register 2 (IRQP2) $1FFF20 + $12

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	PENDING [48:33]															
Write																
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

IRQ Pending Register 3 (IRQP3) $1FFF20 + $13

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	PENDING [64:49]															
Write																
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

IRQ Pending Register 4 (IRQP4) $1FFF20 + $14

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	1	1	1	1	1	1	1	1	1	1	1	PENDING [69:65]				
Write																
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

denotes Reserved Bits

ITCN

Interrupt Control Register (ICTL)

Bits	Name	Description	
15	INT	**Interrupt**	
		0	No interrupt is being presented to the core
		1	An interrupt is being presented to the core
14 - 13	IPLC	**Interrupt Priority Level Core**	
		This bit field reflects the state of the new interrupt priority level bits being presented to the core at the time the last IRQ was taken.	
		00	Required nested exception priority levels are 0, 1, 2, or 3
		01	Required nested exception priority levels are 1, 2, or 3
		10	Required nested exception priority levels are 2 or 3
		11	Required nested exception priority level is 3
12 - 6	VAB	**Vector Number**	
		This field shows bits [7:1] of the Vector Number of the last IRQ. The field is only updated when the core jumps to a new interrupt service routine.	
5	INT_DIS	**Interrupt Disable**	
		Disables all interrupts	
		0	Normal operation (default)
		1	All interrupts disabled

IRQ Control Register (ICTL) $1FFF20 + $1A

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	INT	IPLC		VAB							INT_DIS	0	IRQB_STATE	IRQA_STATE	IRQB_EDG	IRQA_ EDG
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0

denotes Reserved Bits

See the following page for continuation of this register

ITCN

Interrupt Control Register (ICTL) continued

Bits	Name	Description
3	IRQB_STATE	**State of $\overline{\text{IRQB}}$**
		This bit reflects the state of the external IRQB
2	IRQA_STATE	**State of $\overline{\text{IRQA}}$**
		This bit reflects the state of the external **$\overline{\text{IRQA}}$**
1	IRQB_EDG	**$\overline{\text{IRQB}}$ Edge**
		This bit controls whether the external**$\overline{\text{IRQB}}$** interrupt is edge or level sensitive. Automatically level sensitive during Stop and Wait modes.
		0 **$\overline{\text{IRQB}}$** interrupt is level sensitive (default)
		1 **$\overline{\text{IRQB}}$** interrupt is falling edge sensitive
0	IRQA_EDG	**$\overline{\text{IRQA}}$ Edge**
		This bit controls whether the external $\overline{\text{IRQA}}$ interrupt is edge or level sensitive. Automatically level sensitive during Stop and Wait modes.
		0 $\overline{\text{IRQA}}$ interrupt is low level sensitive (default)
		1 $\overline{\text{IRQA}}$ interrupt is falling edge sensitive

IRQ Control Register (ICTL) $1FFF20 + $1A	Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Read	INT	IPLC		VAB							INT_DIS	0	IRQB_STATE	IRQA_STATE	IRQB_EDG	IRQA_ EDG
	Write																
	Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0

denotes Reserved Bits

Application: _____ Date: _____

_____ Programmer: _____

DMA

Source Address High Register (DMASAH)

Bits	Name	Description
7 - 0	DMASAH	**DMA Source Address (High)**
		This is the upper byte of the DMA Source Address. DMA data is read from this address. If post-incrementing (or decrementing) is enabled, this value is updated with each read transaction.

DMA Source Address High Register (DMASAH) Base + $6	Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Read	0	0	0	0	0	0	0	0	DMA SOURCE ADDRESS (HIGH)							
	Write	0	0	0	0	0	0	0	0								
	Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

denotes Reserved Bits

DMA

Source Address Low Register (DMASAL)

Bits	Name	Description
15 - 0	DMASAL	**DMA Source Address (Low)**
		This is the lower word of the DMA Source Address. DMA data is read from this address. If post-incrementing (or decrementing) is enabled, this value is updated with each read transaction.

DMA Source Address Low Register (DMASAL) Base + $5	Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Read / Write	DMA SOURCE ADDRESS (LOW)															
	Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DMA

Destination Address High Register (DMADAH)

Bits	Name	Description
7 - 0	DMADAH	**DMA Destination Address (High)**
		This is the upper byte of the DMA Destination Address. DMA data is written from this address. If post-incrementing (or decrementing) is enabled, this value is updated with each write transaction.

DMA Destination Address High Register (DMADAH) Base + $4	Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Read	0	0	0	0	0	0	0	0	DMA DESTINATION ADDRESS (HIGH)							
	Write																
	Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

denotes Reserved Bits

DMA

Destination Address Low Register (DMADAL)

Bits	Name	Description
15 - 0	DMADAL	**DMA Destination Address (Low)**
		This is the lower byte of the DMA Destination Address. DMA data is written from this address. If post-incrementing (or decrementing) is enabled, this value is updated with each write transaction.

DMA Destination Address Low Register (DMADAL) Base + $3	Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Read / Write	DMA DESTINATION ADDRESS (LOW)															
	Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DMA

DMA Transfer Count Register (DMACNT)

Bits	Name	Description
15 - 0	DMACNT	**DMA Transfer Count**
		This is the count of the number of items (bytes or words) to transfer in the DMA data block.

DMA Transfer Count Register (DMACNT) Base + $2	Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Read	DMA TRANSFER COUNT															
	Write																
	Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DMA

DMA Circular Queue Size Register (DMACQS)

Bits	Name	Description
15 - 0	DMACQS	**DMA Circular Queue Size**
		This is the size of the Circular Queue. The register defaults to zero. When zero, the circular queue operation is disabled. As a non-zero, the queue mode is operational.

DMA Circular Queue Size Register (DMACQS) Base + $1	Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Read	DMA CIRCULAR QUEUE SIZE															
	Write																
	Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DMA

DMA Transfer Control Register (DMATC)

Bits	Name	Description	
15	DMAON	**Direct Memory Access On**	
		This bit enables/disables the DMA operation.	
		0	DMA operation is disabled
		1	DMA operation is enabled
14	INTRON	**Interrupt On**	
		This bit enables/disables the DMA interrupt operation	
		0	DMA interrupt operation is disabled
		1	DMA interrupt operation is enabled
13	INTRPEND	**Interrupt Pending**	
		This is the DMA interrupt pending bit	
		0	DMA interrupt is not pending
		1	DMA interrupt is pending
12	DATASIZE	**Data Size**	
		This bit defines the data size of the DMA transfers	
		0	DMA transfers are performed in bytes (8 bits)
		1	DMA transfers are performed in words (16 bits)

DMA Transfer Control Register (DMATC) Base + $0	Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Read	DMA_ON	INTR_ON	INTR_PEND	DATA_SIZE	0	0	0	PERIPH SEL					SS/M		DS/M	
	Write					0	0	0									
	Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

denotes Reserved Bits

See the following page for continuation of this register

DMA

DMA Transfer Control Register (DMATC) continued

Bits	Name	Description	
8 - 4	PERIPH SEL	**PERIPHERAL SELECT**	
		This bit field selects on of 31 possible DMA peripherals. When the field is equal to 0, operation begins upon enabling DMA operation. When a peripheral-to-peripheral DMA transfer is desired, this field should be set to the slower of the two peripherals thereby preventing data overruns.	
3 - 2	SS/M	**Source Sign/Magnitude**	
		This bit field determines how the DMA Source Address Register gets changed after each DMA read operation.	
		0x	Do not change the address register
		10	Increment the address register by data size amount after each read
		11	Decrement the address register by data size amount after each read
1 - 0	DS/M	**Destination Sign/Magnitude**	
		This bit field determines how the DMA Destination Address Register gets changed after each DMA write operation.	
		0x	Do not change the address register
		10	Increment the address register by data size amount after each read
		11	Decrement the address register by data size amount after each read

DMA Transfer Control Register (DMATC) Base + $0	Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Read	DMA_ON	INTR_ON	INTR_PEND	DATA_SIZE	0	0	0	PERIPH SEL					SS/M		DS/M	
	Write					0	0	0									
	Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

denotes Reserved Bits

SCI

SCI Baud Register (SCIBR)

Bits	Name	Description
12 - 0	SBR	**SCI Baud Register**
		This register may be read at any time. Bits 12 through 0 can be written at any time. (SBR = contents of the baud rate registers, a value from 1 to 8191.)

SCI Baud Register (SCIBR) Base + $0	Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Read	0	0	0	SBR												
	Write																
	Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

denotes Reserved Bits

SCI

SCI Control Register (SCICR)

Bits	Name	Description
15	LOOP	**Loop Select**
		This bit enables loop operation. Loop operation disconnects the RXD pin from the SCI and the transmitter output goes into the receiver input. Both transmitter and receiver must be enabled to use the internal loop function as opposed to single wire operation, requiring only one or the other to be enabled.
		0 Normal operation enabled
		1 Loop operation enabled
14	SWAI	**Stop in Wait Mode**
		This bit disables the SCI in the Wait mode.
		0 SCI enabled in Wait mode
		1 SCI disabled in Wait mode
13	RSRC	**Receiver Source**
		When LOOP = 1, the RSRC bit determines the internal feedback path for the receiver.
		0 Receiver input internally connected to transmitter output
		1 Receiver input connected to TXD pin
12	M	**Data Format Mode**
		The Mode bit determines whether data characters are eight or nine bits long.
		0 One start bit, eight data bits, one stop bit
		1 One start bit, nine data bits, one stop bit
11	WAKE	**Wake up Condition**
		This bit determines which condition wakes up the SCI.
		0 Idle line wake up
		1 Address mark wake up
10	POL	**Polarity**
		This bit determines whether to invert the data as it goes from the transmitter to the TXD pin and from the RXD pin to the receiver. All bits (Start, Data, and Stop) will be inverted as they leave the transmit shift register and before they enter the receive shift register.
		0 Doesn't invert transmit and receive data bits (Normal mode)
		1 Invert transmit and receive data bits (Inverted mode)

SCI Control Register (SCICR) Base + $1	Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Read	LOOP	SWAI	RSRC	M	WAKE	POL	PE	PT	TEIE	TIIE	RFIE	REIE	TE	RE	RWU	SBK
	Write																
	Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

See the following page for continuation of this register

SCI

SCI Control Register (SCICR) continued

Bits	Name	Description
9	PE	**Parity Enable**
		This bit enables the parity function. When enabled, the function replaces the most significant bit of the data character with a parity bit.
		0 Parity function disabled
		1 Parity function enabled
8	PT	**Parity Type**
		This bit determines if the SCI generates and checks for even or odd parity of the data bits.
		0 Even Parity
		1 Odd parity
7	TEIE	**Transmitter Empty Interrupt Enable**
		This bit enables Transmit Data Register Empty (TDRE) flag to generate interrupt requests.
		0 TDRE interrupt requests disabled
		1 TDRE interrupt requests enabled
6	TIIE	**Transmitter Idle Interrupt Enable**
		This bit enables the Transmitter Idle (TIDLE) flag to generate interrupt requests.
		0 TIDLE interrupt requests disabled
		1 TIDLE interrupt requests enabled
5	RFIE	**Receiver Full Interrupt Enable**
		This bit enables the Receive Data Register Full (RDRF) flag, or the Overrun (OR) flag to generate interrupt requests.
		0 RDRF and OR interrupt requests disabled
		1 RDRF and OR interrupt requests enabled

SCI Control Register (SCICR) Base + $1	Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Read	LOOP	SWAI	RSRC	M	WAKE	POL	PE	PT	TEIE	TIIE	RFIE	REIE	TE	RE	RWU	SBK
	Write																
	Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

See the following page for continuation of this register

SCI

SCI Control Register (SCICR) continued

Bits	Name	Description	
4	REIE	**Receiver Error Interrupt Enable**	
		This bit enables Receive Error (RE) flags (NF, PF, FE, an OR) to create interrupt requests.	
		0	Error interrupt requests disabled
		1	Error interrupt requests enabled
3	TE	**Transmitter Enable**	
		This bit enables the SCI transmitter, configuring the TXD pin as the SCI transmitter output.	
		0	Transmitter disabled
		1	Transmitter enabled
2	RE	**Receiver Enable**	
		This bit enables the SCI receiver.	
		0	Receiver disabled
		1	Receiver enabled
1	RWU	**Receiver Wake up**	
		This bit enables the wake up function and inhibits further receiver interrupt requests.	
		0	Standby state
		1	Normal operation
0	SBK	**Send Break**	
		Toggling SBK sends one break character (10 or 11 Logic 0s). As long as SBK is set, the transmitter sends Logic 0s.	
		0	No break characters
		1	Transmit break characters

SCI Control Register (SCICR) Base + $1	Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Read	LOOP	SWAI	RSRC	M	WAKE	POL	PE	PT	TEIE	TIIE	RFIE	REIE	TE	RE	RWU	SBK
	Write																
	Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SCI

SCI Control Register 2 (SCISR2)

Bits	Name	Description	
2	RIIE	**Receiver Idle Interrupt Enable**	
		This bit is used to inform the processor a message has completed when using DMA in a wake-up mode of operation. The receiver is normally configured until a message is detected.	
1	TDE	**Transmitter DMA Enable**	
		Setting this bit allows transmit DMA requests to the DMA controller. Fro transmit DMA operation, this bit must be set and a DMA channel must be enabled to utilize the request.	
		0	Transmit DMA disabled
		1	Transmit DMA enabled
0	RDE	**Receiver DMA Enable**	
		Setting this bit allows received DMA requests to the DMA controller. For receive DMA operation, this bit must be set and a DMA channel must be enabled to utilize the request.	
		0	Receive DMA disabled
		1	Receive DMA enabled

SCI Control Register 2 (SCICR2) Base + $2	Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Read	0	0	0	0	0	0	0	0	0	0	0	0	0	RIIE	TDE	RDE
	Write																
	RESET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

denotes Reserved Bits

SCI

SCI Status Register (SCISR)

Bits	Name	Description	
15	TDRE	**Transmit Data Register Empty**	
		This bit is set when the transmit shift register receives a character from the SCI Data Register (SCIDR). Clear TDRE by reading SCISR with TDRE set and then writing to SCI data register in normal mode or by writing the SCIDR with TDE set.	
		0	No character transferred to transmit shift register
		1	Character transferred to transmit shift register; transmit data register empty
14	TIDLE	**Transmitter Ilde**	
		This bit is set when the TDRE flag is set and not data, preamble, or break character is being transmitted. When TIDLE is set, the TXD pin becomes idle (Logic 1). Clear TIDLE by reading the SCI Status Register (SCISR) with TIDLE set and then writing to the SCI Data Register (SCIDR). TIDLE is not generated when a data character, a preamble, or a break is queued and ready to be sent.	
		0	Transmission in progress
		1	No transmission in progress
13	RDRF	**Receive Data Register Full**	
		This bit is set when the data in the receive shift register transfers to the SCI Data Register (SCIDR) Clear RDRF by reading the SCI Status Register (SCISR) with RDRF set and then reading the SCI data register in normal mode or by reading the SCIDR with RDE set.	
		0	Data not available in SCI data register
		1	Received data available in SCI data register

SCI Status Register (SCISR) Base + $3	Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Read	TDRE	TIDLE	RDRF	RIDLE	OR	NF	FE	PF	0	0	0	0	0	0	0	RAF
	Write																
	Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

See the following page for continuation of this register

denotes Reserved Bits

SCI

SCI Status Register (SCISR) continued

Bits	Name	Description	
12	RIDLE	**Receiver Idle Line**	
		This bit is set when 10 consecutive Logic 1s (if M = 0) or 11 consecutive Logic 1s (if M = 1) appear on the receiver input. Once the RIDLE flag is cleared (the receiver detects a Logic 0), a valid frame must again set the RDRF flag before an idle condition can set the RIDLE flag.	
		0	Receiver input is either active now or has never become active since the RIDLE flag was last cleared
		1	Receiver input has become idle (after receiving a valid frame)
11	OR	**Overrun**	
		This bit is set when software fails to read the SCI Data Register (SCIDR) before the receive shift register receives the next frame. The data in the shift register is lost, but the data already in the SCI data register is not affected. Clear OR by reading the SCI Status Register (SCISR) with OR set and then writing the SCI status register with any value.	
		0	No overrun
		1	Overrun
10	NF	**Noise Flag**	
		This bit is set when the SCI detects noise on the receiver input. The NF bit is set during the same cycle as the RDRF flag but does not get set in the case of an overrun. Clear NF by reading the SCI Status Register (SCISR) and then writing the SCI status register with any value.	
		0	No noise
		1	Noise

SCI Status Register (SCISR) Base + $3	Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Read	TDRE	TIDLE	RDRF	RIDLE	OR	NF	FE	PF	0	0	0	0	0	0	0	RAF
	Write																
	Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

See the following page for continuation of this register

denotes Reserved Bits

SCI

SCI Status Register (SCISR) continued

Bits	Name	Description	
9	FE	**Framing Error**	
		This bit is set when a Logic 0 is accepted as the stop bit. The FE bit is set during the same cycle as the RDRF flag but does not get set in the case of an overrun. FE inhibits further data reception until it is cleared. Clear FE by reading the SCI Status Register (SCISR) with FE set and then writing the SCI status register with any value.	
		0	No framing error
		1	Framing error
8	PF	**Parity Error Flag**	
		This bit is set when the parity enable PE bit is set and the parity of the received data does not match its parity bit. Clear PF by reading the SCI Status Register (SCISR) and then writing the SCI status register with any value.	
		0	No parity error
		1	Parity error
0	RAF	**Receiver Active Flag**	
		This bit is set when the receiver detects a Logic 0 during the RT1 time period of the start bit search. RAF is cleared when the receiver detects false start bits (usually from noise or baud rate mismatch) or when the receiver detects a preamble.	
		0	No reception in progress
		1	Reception in progress

SCI Status Register (SCISR) Base + $3	Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Read	TDRE	TIDLE	RDRF	RIDLE	OR	NF	FE	PF	0	0	0	0	0	0	0	RAF
	Write																
	Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

denotes Reserved Bits

SCI

SCI Data Register (SCIDR)

Bits	Name	Description
8 - 0	Receive Data	Data Received
8 - 0	Transmit Data	Data to be Transmitted

SCI Data Register (SCIDR) Base + $4	Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Read	0	0	0	0	0	0	0	Receive Data								
	Write	0	0	0	0	0	0	0	Transmit Data								
	Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

denotes Reserved Bits

Application: _____ Date: _____

Programmer: _____

SPI

SPI Status and Control Register (SPSCR)

Bits	Name	Description
15 - 13	SPR	**SPI Baud Rate**
		These are read/write bits while in Master mode, selects one of four baud rates.
12	DSO	**Data Shift Order**
		This read/write bit determines whether the MSB or LSB bit is transmitted or received first.
		0 MSB transmitted first (MSB->LSB)
		1 LSB transmitted first (LSB->MSB)
11	ERRIE	**Error Interrupt Enable**
		This read/write bit enables the MODF and OVRF bits to generate interrupt requests. Reset clears the ERRIE bit. ERRIE bit enables both the MODF and OVRF bits to generate a receiver/error interrupt request.
		0 MODF and OVRF cannot generate interrupt requests
		1 MODF and OVRF can generate interrupt requests
10	MODFEN	**Mode Fault Enable**
		This read/write bit when set to one allows the MODF flag to be set. If the MODF flag is set, clearing the MODFEN does not clear the MODF flag.
9	SPRIE	**SPI Receiver Interrupt Enable**
		This read/write bit enables interrupt requests generated by the SPRF bit. The SPRF bit is set when a full data length transfers from the Shift Register to the Receive Data Register.
		0 SPRF interrupt requests disabled
		1 SPRF interrupt requests enabled
8	SPMSTR	**SPI Master**
		This read/write bit selects Master mode operation or slave mode operation.
		0 Slave mode
		1 Master mode

SPI Status and Control Register (SPSCR) $1FFFE8	Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Read	SPR			DSO	ERRIE	MODFEN	SPRIE	SPMSTR	CPOL	CPHA	SPE	SPTIE	SPRF	OVRF	MODF	SPTE
	Write																
	Reset	0	1	1	0	0	0	0	1	0	1	0	0	0	0	0	0

See the following page for continuation of this register

denotes Reserved Bits

SPI

SPI Status and Control Register (SPSCR) continued

Bits	Name	Description
7	CPOL	**Clock Parity**
		This read/write bit determines the logic state of the SCLK pin between transmissions. To transmit data between SPI modules, the SPPI modules musts have identical CPOL values.
		0 Falling edge of SCLK starts transmission
		1 Rising edge of the SCLK starts transmission
6	CPHA	**Clock Phase**
		This read/write bit controls the timing relationship between the serial clock and SPI data. To transmit data between SPI modules, there must be identical CPHA values. When CPHA = 0, the $\overline{SS}$ pin of the Slave SPI module musts be set to Logic 1 between full length data transmissions.
5	SPE	**SPI Enable**
		This read/write bit enables the SPI module. Clearing SPE causes a partial reset of the SPI. When setting/clearing this bit, no other bits in the SPSCR should be changed. Failure to following this statement may result in spurious clocks.
		0 SPI module disabled
		1 SPI module enabled
4	SPTIE	**SPI Transmit Interrupt Enable**
		This read/write bit enables interrupt requests generated by the SPTE bit. SPTE is set when a full data length transfers from the Transmit Data Register to the Shift Register.
		0 SPTE interrupt requests disabled
		1 SPTE interrupt request enabled
3	SPRF	**SPI Receiver Full**
		This *read-only* flag enables interrupt requests generated by the SPTE bit.
		0 Receive Data Register not full
		1 Receive Data Register full

SPI Status and Control Register (SPSCR) $1FFFE8	Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Read	SPR			DSO	ERRIE	MODFEN	SPRIE	SPMSTR	CPOL	CPHA	SPE	SPTIE	SPRF	OVRF	MODF	SPTE
	Write																
	Reset	0	1	1	0	0	0	0	1	0	1	0	0	0	0	0	0

See the following page for continuation of this register

denotes Reserved Bits

SPI

SPI Status and Control Register (SPSCR) continued

Bits	Name	Description	
2	**OVRF**	**Overflow**	
		This *read-only* flag is set if software does not read the data in the Receive Data Register before the next full data enters the Shift Register.	
		0	No overflow
		1	Overflow
1	**MODF**	**Mode Fault**	
		This *read-only* flag is set in a slave SPI if the $\overline{SS}$ pin goes high during a transmission with the MODFEN bit set.	
		0	$\overline{SS}$ pin at appropriate logic level
		1	$\overline{SS}$ pin at inappropriate logic level
0	**SPTE**	**SPI Transmitter Empty**	
		This *read-only* flag is set each time the Transmit Data Register transfers a full data length into the Shift Register.	
		0	Transmit Data Register not empty
		1	Transmit Data Register empty

SPI Status and Control Register (SPSCR) $1FFFE8	Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Read	SPR			DSO	ERRIE	MODFEN	SPRIE	SPMSTR	CPOL	CPHA	SPE	SPTIE	SPRF	OVRF	MODF	SPTE
	Write																
	Reset	0	1	1	0	0	0	0	1	0	1	0	0	0	0	0	1

denotes Reserved Bits

Application: _____ Date: _____

_____ Programmer: _____

SPI

SPI Data Size and Control Register (SPDSCR)

Bits	Name	Description	
15	WOM	**Wired OR Mode**	
		0	Wired OR mode disabled
		1	Wired OR mode enabled
14	TDMAEN	**Transmitter DMA Enable**	
		0	Transmitter DMA mode disabled
		1	Transmitter DMA mode enabled
13	RDMAEN	**Receiver DMA Enable**	
		0	Receiver DMA mode disabled
		1	Receiver DMA mode enabled
3 - 0	TDS	**Transmission Data Size**	
		Detailed transmission data provided in the following table:	

DS3 - DS0	Size of Transmission	DS3 - DS0	Size of Transmission
$0	Not Allowed	$8	9 Bits
$1	2 Bits	$9	10 Bits
$2	3 Bits	$A	11 Bits
$3	4 Bits	$B	12 Bits
$4	5 Bits	$C	13 Bits
$5	6 Bits	$D	14 Bits
$6	7 Bits	$E	15 Bits
$7	8 Bits	$F	16 Bits

SPI Data Size and Control Register (SPDSCR) $1FFFE9	Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Read	WOM	TDMAEN	RDMAEN	0	0	0	0	0	0	0	0	0	TDS			
	Write																
	Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1

denotes Reserved Bits

SPI

SPI Data Receive Register (SPDRR)

Bits	Name	Description
15 - 0	R	**Receive**
		This is a *read-only* data register. Reading data from the register will show the last full data received after a complete transmission. The SPRF bit will set when new data transfers to this register.

SPI Data Receive Register (SPDRR) $1FFFEA	Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Read	R15	R14	R13	R12	R11	R10	R9	R8	R7	R6	R5	R4	R3	R2	R1	R0
	Write																
	Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

denotes Reserved Bits

SPI

SPI Data Transmit Register (SPDTR)

Bits	Name	Description
15 - 0	**T**	**Transmit**
		This is a *write-only* data register. Writing data to the register modifies data to the transmit data buffer. When the SPTE bit is set, new data should be written to this register. If new data is not written while in the Master mode, a new transaction will not be initiated until this register is written. When in the Slave mode, the old data will be re-transmitted. All data should be written with the LSB at bit 0.

SPI Data Transmit Register (SPDTR) $1FFFEB	Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Read	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Write	T15	T14	T13	T12	T11	T10	T9	T8	T7	T6	T5	T4	T3	T2	T1	T0
	Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

denotes Reserved Bits

Application: _____ Date: _____

_____ Programmer: _____

ESSI

ESSI Transmit Data Register 0 - 2 (STX0-2)

Bits	Name	Description
15 -0	**DATA**	**Write Data**
		STX0-2 are 16-bit write-only data registers.

ESSI Transmit Data Register (STX0) Base + $0	Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Read																
	Write	DATA															
	Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESSI Transmit Data Register (STX1) Base + $1	Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Read																
	Write	DATA															
	Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESSI Transmit Data Register (STX2) Base + $2	Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Read																
	Write	DATA															
	Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

denotes Reserved Bits

ESSI

ESSI Receive Data Register (SRX)

Bits	Name	Description
15 - 8	HIGH BYTE	SRX is a read-only register. The register always accepts data from the Receiver Shift Register as it becomes full.
7 - 0	LOW BYTE	

ESSI Receive Data Register (SRX) Base + $3	Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Read	HIGH BYTE								LOW BYTE							
	Write																
	Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

denotes Reserved Bits

ESSI

ESSI Status Register (SSR)

Bits	Name	Description
15	IF1	**Input Flag 1**
		This bit is enabled only when SC1 is configured as an input flag and Synchronous mode is selected. If not enabled, the IF1 bit is cleared.
14	IF0	**Input Flag 0**
		This bit is enabled only when SC0 is configured as an input flag and the Synchronous mode is selected. If it is not enabled, the IF0 bit is cleared.
12	TFF	**Transmit FIFO Full**
		This status bit allows monitoring when the TTF is full.
		0 Transmit FIFO can accept more data
		1 Transmit FIFO is full
11	TLS	**Transmit Last Slot**
		This status bit indicates timing of the last transmit slot during the Network mode operation
		0 Not currently transmitting the last time slot of the transmit frame
		1 Last slot of the transmit frame is currently being transmitted
10	RLS	**Receive Last Slot**
		This status bit indicates the timing of the last receive slot during the Network mode operation
		0 Not currently receiving the last time slot of the receive frame
		1 Last slot of the receive frame is currently being received
9	TF1ERR	**Transmit FIFO 1 Error**
		When the transmitter status control TXSF1 is set and the FIFOs are in use, this status bit will indicate the state of FIFO1 is not the same as FIFO0.
		0 State of TXFIFO0 and TXFIFO1 contains the same amount of data
		1 State of TXFIFO is different than the state of the TXFIFO1
8	TF2ERR	**Transmit FIFO 2 Error**
		When the transmitter status control TXSF2 is set and FIFOs are in use, this status bit indicates the state of FIFO2 is not the same as FIFO0.
		0 Status of TXFIFO2 matches the other enabled TXFIFOs
		1 Status (data content level) of TXFIFO2 is different than the other enabled TXFIFOs

ESSI Status Register (SSR) Base + $4	Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Read	IF1	IF0		TFF	TLS	RLS	TF1ERR	TF2ERR	RDR	TDE	ROE	TUE	TFS	RFS	RFF	TFE
	Write																
	Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

denotes Reserved Bits

See the following page for continuation of this register

ESSI

ESSI Status Register (SSR) continued

Bits	Name	Description
7	RDR	**Receive Data Ready Flag**
		This flag bit is set when Receive Data Register or the receive FIFO is loaded with a new value. The RDR is cleared when the CPU reads the SRX register. When RXFIFO is enabled, RDR is cleared when receive FIFO is empty.
6	TDE	**Transmit Data Register Empty**
		This flag bit is set when there is no data waiting to be transferred to the TXSR register. A transmit FIFO is enabled when there is at least one empty slot in STX or TXFIFO. When the TXFIFO is not enabled, the STX is empty.
5	ROE	**Receive Overrun Error**
		This flag bit is set when the Receive Shift Register (RXSR) is enabled, filled, ready to transfer to the SRX or the RXFIFO registers, and when these registers are already full.
4	TUE	**Transmitter Underrun Error**
		This flag bit is set when the TXSR is empty, or when there is no data to be transmitted, as indicated by the TDE bit being set, and a transmit time slot occurs. When a Transmit Underrun Error occurs, the previously sent data is retransmitted.
3	TFS	**Transmit Frame Sync**
		When set, this flag bit indicates a frame sync occurred during transmission of the last word written to the STX register.
2	RFS	**Receive Frame Sync**
		When set, this flag bit indicates a frame sync occurred during receiving of the next word into the SRX register.
1	RFF	**Receive FIFO Full**
		When set, this bit indicates data can be read using the SRX register. The data level in the RXFIFO must reach the your selected Receive FIFO Watermark (RFWM) threshold.
0	TFE	**Transmit FIFO Empty**
		When set, the TFE bit indicates data can be written to the TXFIFO register. The TFE bit is cleared by writing data to the STX register until the TXFIFO data content level reaches the watermark level.

ESSI Status Register (SSR) Base + $4	Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Read	IF1	IF0		TFF	TLS	RLS	TF1ERR	TF2ERR	RDR	TDE	ROE	TUE	TFS	RFS	RFF	TFE
	Write																
	Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

denotes Reserved Bits

ESSI

ESSI Control Register 2 (SCR2)

Bits	Name	Description	
15	RIE	**Receive Interrupt Enable**	
		This control bit allows interrupting the program controller	
		0	Receive FIFO is disabled: No interrupt is generated
		1	Receive FIFO is disabled: Interrupt is generated if the RDR flag in the SSR is set
		0	Receive FIFO is enabled: No interrupt is generated
		1	Receive FIFO is enabled: Interrupt is generated if the RFF flag in the SSR is set
14	TIE	**Transmit Interrupt Enable**	
		This control bit provides program controller interruption	
		0	Transmit FIFO is disabled: No interrupt is generated
		1	Transmit FIFO is disabled: Interrupt is generated if the TDE flag in the SSR is set
		0	Transmit FIFO is enabled: No interrupt is generated
		1	Transmit FIFO is enabled: Interrupt is generated if the TFE flag in the SSR is set
13	RE	**Receive Enable**	
		This control bit enables the receive portion of the ESSI	
		0	Receiver is disabled by inhibiting data transfer into the SRX
		1	Receiver portion of the ESSI is enabled and received data will be processed beginning with the next frame sync
12	TE0	**Transmit Enable 0**	
		This control bit enables the transfer of the contents of the STX0 register to its TXSR0	
		0	The transmitter continues to send data in TXSRO, then transmitter is disabled
		1	On the next frame boundary, the transmit 0 portion of the ESSI is enabled
11	TE1	**Transmit Enable 1**	
		This control bit enables the transfer of the contents of the STX1 register to its TXSR1	
		0	Transmitter continues to send data currently in TXSR1, then disables transmitter
		1	On the next frame boundary, the transmit 1 portion of the ESSI is enabled
10	TE2	**Transmit Enable 2**	
		This control bit enables the transfer of the contents of the STX2 register to its TXSR2	
		0	Transmitter continues to send data currently in TXSR2, then disables transmitter
		1	On the next frame boundary, the transmit 2 portion of the ESSI is enabled

ESSI Control Register 2 (SCR2) Base + $5	Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Read	RIE	TIE	RE	TE0	TE1	TE2			SYN	TSHFD	TSCKP	ESSIEN	NET	TFSI	TFSL	TEFS
	Write																
	Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

denotes Reserved Bits

See the following page for continuation of this register

ESSI

ESSI Control Register 2 (SCR2) continued

Bits	Name	Description
7	SYN	**Synchronous Mode**
		This control bit enables the Synchronous mode of operation
6	TSHFD	**Transmit Shift Direction**
		This bit controls whether the MSB or LSB is transmitted first for the transmit section
		0 MSB is transmitted first
		1 LSB is transmitted first
5	TSCKP	**Transmit Clock Parity**
		This control bit determines which bit clock edge is used to clock out data in the transmit
		0 Data is clocked out on the rising edge of the bit clock
		1 Falling edge of the bit clock is used to clock the data out
4	ESSIEN	**ESSI Enable**
		This control bit enables and disables the ESSI
		0 The ESSI is disabled
		1 The ESSI is enabled
3	NET	**Network Mode**
		This control bit selects the operational mode of the ESSI
		0 Normal mode is selected
		1 Network mode is selected
2	TFSI	**Transmit Frame Sync Invert**
		This control bit selects the logic of frame sync I/O
		0 Frame sync is active high
		1 Frame sync is active low
1	TFSL	**Transmit Frame Sync Length**
		This control bit selects the length of the frame sync signal to be generated or recognized
		0 A one-word long frame sync is selected
		1 A one-bit long frame sync is selected
0	TEFS	**Transmit Early Frame Sync**
		This bit controls when the frame sync is initiated for the transmit and receive sections
		0 Frame sync is initiated as the first bit of data is transmitted
		1 Frame sync is initiated one bit before the data is transmitter

ESSI Control Register 2 (SCR2) Base + $5	Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Read	RIE	TIE	RE	TE0	TE1	TE2			SYN	TSHFD	TSCKP	ESSIEN	NET	TFSI	TFSL	TEFS
	Write																
	Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

denotes Reserved Bits

ESSI

ESSI Control Register 3 (SCR3)

Bits	Name	Description	
15	DIV4DIS	**Divider 4 Disable**	
		0	FIX_CLK is equal to the IP_CLK/4 for both transmitter and receiver clock generator circuits
		1	FIX_CLK is equal to the IP_CLK
14	RSHFD	**Receive Shift Direction**	
		This control bit determines if MSB or LSB is received first for the receive section	
		0	Data received MSB first
		1	Data received LSB first
13	RSCKP	**Receive Clock Polarity**	
		This control bit determines if bit clock edge is used to capture data for the receive section	
		0	Captured data during clock falling edge
		1	Capture data input during rising edge of the clock
12	RDMAE	**Receive DMA Enable**	
		This bit allows the ESSI to request a DMA module transfer of received data	
		0	If receive FIFO is disabled: No DMA transfer is requested
		1	If receive FIFO is disabled: DMA request is generated when the ESSI RDR bit is set
		0	If receive FIFO is enabled: No DMA transfer is requested
		1	If receive FIFO is enabled: DMA request is generated when the ESSI FRR bit in SSR is set
11	TDMAE	**Transmit DMA Enable**	
		This bit allows the ESSI to request a DMA transfer for the next data to transmit	
		0	If transmit FIFO is disabled: No DMA transfer is requested
		1	If transmit FIFO is disabled: DMA request is generated when the ESSI TDE bit in TDE is set
		0	If transmit FIFO is enabled: No DMA transfer is requested
		1	If transmit FIFO is enabled: DMA request is generated when the ESSI TFE bit in SSR is set
10	RFSI	**Receive Frame Sync Invert**	
		This bit selects the logic of frame sync I/O for the receive section	
		0	Frame sync is active high
		1	Frame sync is active low

ESSI Control Register 3 (SCR3) Base + $6	Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Read	DIV4DIS	RSHFD	RSCKP	RDMAE	TDMAE	RFSI	RFSL	REFS	RFEN	TFEN	INIT			SYN RST	RLIE	TLIE
	Write																
	Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

denotes Reserved Bits

See the following page for continuation of this register

ESSI

ESSI Control Register 3 (SCR3) continued

Bits	Name	Description	
2	SYNRST	**Frame Sync Reset**	
		0	Data must be read to be cleared from the registers
		1	Resets the accumulation of data in the SRX register and RXFIFO bit in frame sync
1	RLIE	**Receive Last Slot Interrupt Enable**	
		0	The Receive Last Slot Interrupt is disabled
		1	An interrupt is generated after the last slot of a receive frame ends when the ESSI is in the Network mode. Interrupt occurs regardless of the receive mask register setting.
0	TLIE	**Transmit Last Slot Interrupt Enable**	
		0	The Transmit Last Slot Interrupt is disabled
		1	An interrupt is generated at the beginning of the last slot of the transmit frame when the ESSI is in the Network mode. The interrupt occurs regardless of the transmit mask register setting.

ESSI Control Register 3 (SCR3) Base + $6	Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Read	DIV4DIS	RSHFD	RSCKP	RDMAE	TDMAE	RFSI	RFSL	REFS	RFEN	TFEN	INIT			SYN RST	RLIE	TLIE
	Write																
	Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

denotes Reserved Bits

ESSI

ESSI Control Register 4 (SCR4)

Bits	Name	Description	
10	TXSFO	**Transmit Status Flag Control 0**	
		0	The indicated transmitter does not affect the status bits
		1	The indicated transmitter status will affect the TFF, TDE, TUE, and TFE status bits
9	TXSF1	**Transmit Status Flag Control 1**	
		0	The indicated transmitter does not affect the status bits
		1	The indicated transmitter status will affect the TFF, TDE, TUE, and TFE status bits
8	TXSF2	**Transmit Status Flag Control 2**	
		0	The indicated transmitter does not affect the status bits
		1	The indicated transmitter status will affect the TFF, TDE, TUE, and TFE status bits
6	SSC1	**Select SC1**	
		This bit controls the functionality of the SC1 signal and is used only if SYN = 1 and TE2 = 0	
		0	The SC1 acts as the serial I/O flag (IF1/OF1)
		1	When configured as an output (SCD1 = 1), the SC1 signal acts as the transmitter 0 driver-enable, enabling an external buffer for the transmitter 0 output
5	SCKD	**Clock Source Direction**	
		This bit configures the source of the clock signal used to clock the transmit shift register in Asynchronous mode and both Transmit and Receive Shift Registers are in Synchronous mode	
		0	SCK pin is configured as an input. This pin is used as the clock source
		1	Internal transmit clock generator is selected. This clock is output on the SCK pin
4	SCD2	**Serial Control Direction 2**	
		This bit controls the direction of the SC2 pin	
		0	SC2 is an input
		1	SC2 is an output

ESSI Control Register 4 (SCR4) Base + $7	Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Read						TXSFO	TXSF1	TXSF2		SSC1	SCKD	SCD2	SCD1	SCD0	OF1	OF0
	Write																
	Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

denotes Reserved Bits

See the following page for continuation of this register

ESSI

ESSI Control Register 4 (SCR4) continued

Bits	Name	Description	
3	SCD1	**Serial Control Direction 1**	
		This bit controls the direction of the SC1 pin depending on the stat of the other controls	
		0	SC1 is an input
		1	SC1 is an output
2	SCD0	**Serial Control Direction 0**	
		This bit controls the direction of the SC0 pin depending on the state of other controls	
		0	SC0 is an input
		1	SC0 is an output
1	OF1	**Serial Output Flag 1**	
		When SC1 is configured for Output Flag functionality the control bit determines the state of the output pin. Changes in the OF1 bit will appear on the SC1 pin at the beginning of the next frame in the Normal mode, or at the beginning of the next time slot in the Network mode.	
0	OF0	**Serial Output Flag 0**	
		When SC0 is configured for Output Flag functionality, the control bit determines the state of the output pin. Changes in the OF0 bit will appear on the SC0 pin at the beginning of the next frame in the Normal mode, or at the beginning of the next time slot in the Network mode.	

ESSI Control Register 4 (SCR4) Base + $7	Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Read						TXSFO	TXSF1	TXSF2		SSC1	SCKD	SCD2	SCD1	SCD0	OF1	OF0
	Write																
	Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

denotes Reserved Bits

ESSI

ESSI Transmit and Receive Control Registers (STXCR, SRXCR)

Bits	Name	Description
15	PSR	**Prescaler Range**
		This bit controls a fixed divide-by-eight prescaler in series with the variable prescaler. It extends the range of the prescaler for those cases where a slower bit clock is desired.
		0 When the PSR bit is cleared, the fixed prescaler is by-passed
		1 When the PSR bit is set, the fixed divide-by eight prescaler is operational
14 - 13	WL	**Word Length Control**
		These bits are used to select the length of the data words being transferred by the ESSI
12 - 8	DC	**Frame Rate Divider Control**
		This bit field controls the divide ratio for the programmable frame rate dividers
7 - 0	PM	**Prescaler Modulus Select**
		This bit field specifies the divide ratio of the prescale divider in the ESSI clock generator. This prescaler is used only in internal clock mode to divide the internal peripheral clock.

ESSI Transmit Control Register (STXCR) Base + $8	Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Read	PSR	WL		DC					PM							
	Write																
	Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESSI Receive Control Register (SRXCR) Base + $9	Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Read	PSR	WL		DC					PM							
	Write																
	Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ESSI

ESSI Time Slot Register (STSR)

Bits	Name	Description
15 - 0	STSR	**ISSI Time Slot Register**
		Used when data is not to be transmitted in an available transmit time slot. The time slot register is a write-only register behaving like an alternative transmit data register, except instead of transmitting data, the STD pin signal is tri-stated. Using this register is important in avoiding overflow/underflow during inactive time slots.

ESSI Time Slot Register (STSR) Base + $A	Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Read																
	Write	DUMMY REGISTER, WRITTEN DURING INACTIVE TIME SLOTS (NETWORK MODE)															
	Reset	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

denotes Reserved Bits

ESSI

ESSI FIFO Control/Status Register (SFCSR)

Bits	Name	Description	
15 - 12	**RFCNT**	**Receive FIFO Counter**	
		This read-only bit field indicates the number of data words in the RXFIFO. Following demonstrates the RFCNT bit field encoding.	
		0000	0 Data words in RXFIFO
		0001	1 Data word in RXFIFO
		0010	2 Data words in RXFIFO
		0011	3 Data words in RXFIFO
		0100	4 Data words in RXFIFO
		0101	5 Data words in RXFIFO
		0110	6 Data words in RXFIFO
		0111	7 Data words in RXFIFO
		1000	8 Data words in RXFIFO
11 - 8	**TFCNT**	**Transmit FIFO Counter**	
		This read-only bit field indicates the number of data words in the TXFIFO. Following demonstrates the TFCNT bit field encoding.	
		0000	0 Data words in TXFIFO
		0001	1 Data word in TXFIFO
		0010	2 Data words in TXFIFO
		0011	3 Data words in TXFIFO
		0100	4 Data words in TXFIFO
		0101	5 Data words in TXFIFO
		0110	6 Data words in TXFIFO
		0111	7 Data words in TXFIFO
		1000	8 Data words in TXFIFO

ESSI FIFO Control/Status Register (SFCSR) Base + $B	Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Read	RFCNT				TFCNT				RFWM				TFWM			
	Write																
	Reset	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1

denotes Reserved Bits

See the following page for continuation of this register

ESSI

ESSI FIFO Control/Status Register (SFCSR) continued

7 - 4	RFWM	Receive FIFO Full Watermark	
		This bit field controls the threshold setting of the Receive FIFO Full Flag will be set. RFF is set whenever the data level in the RXFIFO reaches the selected threshold. The following data shows the status of RFF for all data levels of the RXFIFO.	
		0000	Reserved
		0001	RFF set when at least 1 data word has been written to the RXFIFO. Set when RXFIFO = 1, 2, 3, 4, 5, 6, 7, or 8 data words
		0010	RFF set when 2 or more data word has been written to the RXFIFO. Set when RXFIFO = 2, 3, 4, 5, 6, 7, or 8 data words
		0011	RFF set when 3 or more data word has been written to the RXFIFO. Set when RXFIFO = 3, 4, 5, 6, 7, or 8 data words
		0100	RFF set when 4 or more data word has been written to the RXFIFO. Set when RXFIFO = 4, 5, 6, 7, or 8 data words
		0101	RFF set when 5 or more data word has been written to the RXFIFO. Set when RXFIFO = 5, 6, 7, or 8 data words
		0110	RFF set when 6 or more data word has been written to the RXFIFO. Set when RXFIFO = 6, 7, or 8 data words
		0111	RFF set when 7 or more data word has been written to the RXFIFO. Set when RXFIFO = 7, or 8 data words
		1000	RFF set when 8 data word has been written to the RXFIFO. Set when RXFIFO = 8 data words
3 - 0	TDWM	Transmit FIFO Empty Watermark	
		This bit field controls the threshold where the Transmit FIFO Empty flag is set. The following data provides the status of TFE for all data levels of the TXFIFO.	
		0000	Reserved
		0001	TFE set when there is 1 empty slot in TXFIFO. (Default) Transmit FIFO empty is set when TXFIFO = <7 data
		0010	TFE set when there is 2 or more empty slots in TXFIFO. Transmit FIFO empty is set when TXFIFO = <6 data
		0011	TFE set when there is 3 empty slot in TXFIFO. Transmit FIFO empty is set when TXFIFO = <5 data
		0100	TFE set when there is 4 empty slot in TXFIFO. Transmit FIFO empty is set when TXFIFO = <4 data
		0101	TFE set when there is 5 empty slot in TXFIFO. Transmit FIFO empty is set when TXFIFO = <3 data
		0110	TFE set when there is 6 empty slot in TXFIFO. Transmit FIFO empty is set when TXFIFO = <2 data
		0111	TFE set when there is 7 empty slot in TXFIFO. Transmit FIFO empty is set when TXFIFO = <1 data
		1000	TFE set when there is 8 empty slot in TXFIFO. Transmit FIFO empty is set when TXFIFO = <0 data

ESSI FIFO Control/Status Register (SFCSR) Base + $B	Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Read	RFCNT				TFCNT				RFWM				TFWM			
	Write																
	Reset	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1

denotes Reserved Bits

ESSI

ESSI Transmit Slot Mask Registers (TSMA, TSMB)

15 - 0	TSMA	Transmit Slot Mask A	
		This is a read/write register. In the Network mode, these registers are used by the transmitter to determine which action to take in the current transmission slot. Depending on the setting of the bits, the transmitter either tri-states the transmitter data signal or transmits a data word and generates the appropriate transmit status. TSMA and TSMB can be viewed as a single 32-bit register named TSM. Bit n in TSM (TSMn) is an enable/disable control bit for transmission in slot N.	
		0	The transmit data signal of the transmitter is tri-stated during transmit time slot N. Data is not transferred to the TXSR, therefore transmit status flags are not changed
		1	The transmit sequence proceeds normally. Data is transferred from the STX register (or TXFIFO, if enabled) to the shift register during slot N. Appropriate flags are set
16 - 31	TSMB	Transmit Slot Mask B	
		This is a read/write register. In the Network mode, these registers are used by the transmitter to determine which action to take in the current transmission slot. Depending on the setting of the bits, the transmitter either tri-states the transmitter data signal or transmits a data word and generates the appropriate transmit status. TSMA and TSMB can be viewed as a single 32-bit register.	
		0	The transmit data signal of the transmitter is tri-stated during transmit time slot N. Data is not transferred to the TXSR, therefore transmit status flags are not changed
		1	The transmit sequence proceeds normally. Data is transferred from the STX register (or TXFIFO, if enabled) to the shift register during slot N. Appropriate flags are set

ESSI Transmit Slot Mask Register (TSMA) Base + $C	Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Read	TSMA [15:0]															
	Write																
	Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

ESSI Transmit Slot Mask Register (TSMB) Base + $D	Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Read	TSMB [31:16]															
	Write																
	Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

ESSI

ESSI Receive Slot Mask Registers (RSMA, RSMB)

15 - 0	RSMA	Receive Slot Mask A	
		The Receive Slot Mask Registers are two 16-bit read/write registers. In the Network mode, these registers are used by the receiver to determine which action to take in the current time slot. Depending on the setting of the bits, the receiver either ignores the receiver data signal(s), or receives a data word, generating the appropriate receive status. RSMA and RSMB can be viewed as one 32-bit register, RSM. Bit n in RSM (RSMn) is an enable/disable control bit for time slot N.	
		0	Data is not transferred from the Receive Shift Register (RXSR) to the Receive Data (SRX) register, therefore the RDR and ROE flags are not set
		1	The receive sequence proceeds normally. Data is received during slot N, and the RDR flag is set
31 - 16	RSMB	Receive Slot Mask B	
		The Receive Slot Mask Registers are two 16-bit read/write registers. In the Network mode, these registers are used by the receiver to determine which action to take in the current time slot. Depending on the setting of the bits, the receiver either ignores the receiver data signal(s), or receives a data word, generating the appropriate receive status. RSMA and RSMB can be viewed as one 32-bit register, RSM. Bit n in RSM (RSMn) is an enable/disable control bit for time slot N.	
		0	Data is not transferred from the Receive Shift Register (RXSR) to the Receive Data (SRX) register, therefore the RDR and ROE flags are not set
		1	The receive sequence proceeds normally. Data is received during slot N, and the RDR flag is set

ESSI Receive Slot Mask Register (RSMA) Base + $E	Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Read	RSMA [15:0]															
	Write																
	Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

ESSI Receive Slot Mask Register (RSMB) Base + $F	Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Read	RSMB [31:16]															
	Write																
	Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

TMR

TMR Compare Register 1 (CMP1)

Bits	Name	Description
15 - 0	COMPARISON VALUE1	**Timer Compare Register 1**
		These read/write registers store the value used for comparison with counter value.

TMRA0_CMP1 (Timer A, Channel 0 Compare 1)—Address: TMRA_BASE + $0
TMRA1_CMP1 (Timer A, Channel 1 Compare 1)—Address: TMRA_BASE + $8
TMRA2_CMP1 (Timer A, Channel 2 Compare 1)—Address: TMRA_BASE + $10
TMRA3_CMP1 (Timer A, Channel 3 Compare 1)—Address: TMRA_BASE + $18

TMR Compare Register1 (CMP1) $1FFE80 +$0, $8, $10, $18	Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Read	COMPARISON VALUE 1															
	Write																
	Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TMR

TMR Compare Register 2 (CMP2)

Bits	Name	Description
15 - 0	COMPARISON VALUE2	**Timer Compare Register 2**
		These read/write registers store the value used for comparison with counter value.

TMRA0_CMP2 (Timer A, Channel 0 Compare 2)—Address: TMRA_BASE + $1
TMRA1_CMP2 (Timer A, Channel 1 Compare 2)—Address: TMRA_BASE + $9
TMRA2_CMP2 (Timer A, Channel 2 Compare 2)—Address: TMRA_BASE + $11
TMRA3_CMP2 (Timer A, Channel 3 Compare 2)—Address: TMRA_BASE + $19

TMR Compare Register (CMP2) $1FFE80 +$1, $9, $11, $19	Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Read	COMPARISON VALUE 2															
	Write																
	Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TMR

TMR Capture Register (CAP)

Bits	Name	Description
15 - 0	CAPTURE VALUE	**Timer Capture Register**
		These read/write registers store the value captured from the counter.

TMRA0_CAP (Timer A, Channel 0 Capture)—Address: TMRA_BASE + $2
TMRA1_CAP (Timer A, Channel 1 Capture)—Address: TMRA_BASE + $A
TMRA2_CAP (Timer A, Channel 2 Capture—Address: TMRA_BASE + $12
TMRA3_CAP (Timer A, Channel 3 Capture)—Address: TMRA_BASE + $1A

TMR Capture Register (CAP) $1FFE80 + $2, $A, $12, $1A	Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Read	CAPTURE VALUE															
	Write																
	Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TMR

TMR Load Register (LOAD)

Bits	Name	Description
15 - 0	LOAD VALUE	**Timer Load Register**
		These read/write registers store the value used to load the counter

TMRA0_LOAD (Timer A, Channel 0 Load)—Address: TMRA_BASE + $3
TMRA1_LOAD (Timer A, Channel 1 Load)—Address: TMRA_BASE + $B
TMRA2_LOAD (Timer A, Channel 2 Load)—Address: TMRA_BASE + $13
TMRA3_LOAD (Timer A, Channel 3 Load)—Address: TMRA_BASE + $1B

TMR Load Register (LOAD) $1FFE80 + $3, $B, $13, $1B	Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Read	LOAD VALUE															
	Write																
	Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TMR

TMR Hold Register (HOLD)

Bits	Name	Description
15 - 0	HOLD VALUE	**Timer Hold Register**
		These read/write registers store the channel's value whenever any counter is read.

TMRA0_HOLD (Timer A, Channel 0 Hold)—Address: TMRA_BASE + $4
TMRA1_HOLD (Timer A, Channel 1 Hold)—Address: TMRA_BASE + $C
TMRA2_HOLD (Timer A, Channel 2 Hold)—Address: TMRA_BASE + $14
TMRA3_HOLD (Timer A, Channel 3 Hold)—Address: TMRA_BASE + $1C

TMR Hold Register (HOLD) $1FFE80 + $4, $C, $14, $1C	Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Read	HOLD VALUE															
	Write																
	Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

denotes Reserved Bits

TMR

TMR Counter Register (CNTR)

Bits	Name	Description
15 - 0	COUNTER	Timer Counter Register
		These read/write registers are the counters.

TMRA0_CNTR (Timer A, Channel 0 Counter)—Address: TMRA_BASE + $5
TMRA1_CNTR (Timer A, Channel 1 Counter)—Address: TMRA_BASE + $D
TMRA2_CNTR (Timer A, Channel 2 Counter)—Address: TMRA_BASE + $15
TMRA3_CNTR (Timer A, Channel 3 Counter)—Address: TMRA_BASE + $1D

TMR Counter Register (CNTR) $1FFE80 + $5, $D, $15, $1D	Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Read	COUNTER															
	Write																
	Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

denotes Reserved Bits

TMR

TMR Control Register (CTL)

Bits	Name	Description	
15 - 13	CM	**Count Mode**	
		These bits control the basic counting behavior of the counter	
		000	No operation
		001	Count rising edges of primary source.
		010	Count rising and falling edges of primary source.
		011	Count rising edges of primary source while secondary input is active high
		100	Quadrature Count mode uses primary and secondary sources
		101	Count primary source rising edges, secondary source specifies direction (1=minus)
		110	Edge of secondary source triggers primary count until compared
		111	Cascaded Counter mode (up/down)
12 - 9	PCS	**Primary Count Service Source**	
		These bits select the primary count source	
		0000	Counter 0 input pin (TIO0)
		0001	Counter 1 input pin (TIO1)
		0010	Counter 2 input pin (TIO2)
		0011	Counter 3 input pin (TIO3)
		0100	Counter 0 output (OFLAG0)
		0101	Counter 1 output (OFLAG1)
		0110	Counter 2 output (OFLAG2)
		0111	Counter 3 output (OFLAG3)
		1000	Prescaler (IPBus clock divide by 1)
		1001	Prescaler (IPBus clock divide by 2)
		1010	Prescaler (IPBus clock divide by 4)
		1011	Prescaler (IPBus clock divide by 8)
		1100	Prescaler (IPBus clock divide by 16)
		1101	Prescaler (IPBus clock divide by 32)
		1110	Prescaler (IPBus clock divide by 64)
		1111	Prescaler (IPBus clock divide by 128)

TMR Control Register (CTL) $1FFE80 + $6, $E, $16, $1E	Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Read	CM			PCS				SCS		ONCE	LENGTH	DIR	EXT INIT	OM (OFLAG)		
	Write																
	Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

See the following page for continuation of this register

TMR

TMR Control Register (CTL) continued

Bits	Name	Description	
8 - 7	SCS	**Secondary Source**	
		This bit field provides additional information used for counting, such as direction.	
		00	Counter 0 input pin (TIO0)
		01	Counter 1 input pin (TIO1)
		10	Counter 2 input pin (TIO2)
		11	Counter 3 input pin (TIO3)
6	ONCE	**Count Once**	
		This bit selects continuous or one-shot counting mode.	
		0	Count repeatedly
		1	Count until compare, then stop. Counting up: compares when counter reaches CMP1 value. Counting down: compares when counter reaches CMP2 value.
5	LENGTH	**Count Length**	
		Determines whether counter counts to the compare value, reinitializing itself.	
		0	Roll-over
		1	Count until compare, then reinitialize

TMRA0_CTRL (Timer A, Channel 0 Control)—Address: TMRA_BASE + $6
TMRA1_CTRL (Timer A, Channel 1 Control)—Address: TMRA_BASE + $E
TMRA2_CTRL (Timer A, Channel 2 Control)—Address: TMRA_BASE + $16
TMRA3_CTRL (Timer A, Channel 3 Control)—Address: TMRA_BASE + $1E

TMR Control Register (CTL) $1FFE80 + $6, $E, $16, $1E	Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Read	CM			PCS				SCS		ONCE	LENGTH	DIR	EXT INIT	OM (OFLAG)		
	Write																
	Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

See the following page for continuation of this register

TMR

TMR Control Register (CTL) continued

Bits	Name	Description	
4	DIR	**Direction**	
		This bit selects either normal count up direction, or the reverse count down direction.	
		0	Count up
		1	Count down
3	EXTINIT	**External Initialization**	
		This bit enables another counter/timer within the same module to force the reinitialization of this counter/timer when the other counter has an active compare event	
		0	External counter/timers cannot force a reinitialization of this counter/timer
		1	External counter/timers may force reinitialization of this counter/timer
2 - 0	OM	**Output mode**	
		This bit field determines the mode of operation for the OFLAG output signal.	
		000	Asserted while counter is active
		001	Clear OFLAG output on successful compare
		010	Set OFLAG output on successful compare
		011	Toggle OFLAG output on successful compare
		100	Toggle OFLAG output using alternating compare registers
		101	Set on compare, cleared on secondary source input edge
		110	Set on compare, cleared on counter roll over
		111	Enable Gated Clock output while counter is active

TMRA0_CTRL (Timer A, Channel 0 Control)—Address: TMRA_BASE + $6
TMRA1_CTRL (Timer A, Channel 1 Control)—Address: TMRA_BASE + $E
TMRA2_CTRL (Timer A, Channel 2 Control)—Address: TMRA_BASE + $16
TMRA3_CTRL (Timer A, Channel 3 Control)—Address: TMRA_BASE + $1E

TMR Control Register (CTL) $1FFE80 + $6, $E, $16, $1E	Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Read	CM			PCS				SCS		ONCE	LENGTH	DIR	EXT INIT	OM (OFLAG)		
	Write																
	Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TMR

TMR Status/Control Registers (SCR)

Bits	Name	Description
15	TCF	**Timer Compare Flag**
		This bit is set when a successful compare occurs. Cleared by writing a 0 to it.
14	TCFIE	**Timer Compare Flag Interrupt Enable**
		When set, this bit enables interrupts when the TCF bit is set.
13	TOF	**Timer Overflow Flag**
		This bit is set when the counter rolls over its maximum value $FFFF or $0000, depending on count direction. Clear the bit by writing 0 to it.
12	TOFIE	**Timer Overflow Flag Interrupt Enable**
		When set, this bit enables interrupts when the TOF bit is set.
11	IEF	**Input Edge Flag**
		This bit is set when a positive input transition occurs. Clear the bit by writing 0 to it.
10	IEFIE	**Input Edge Flag Interrupt Enable**
		When set, this bit enables interrupts when the IEF bit is set.
9	IPS	**Input Polarity Select**
		When set, this bit inverts the input signal polarity.
8	INPUT	**External Input Signal**
		This read-only bit reflects the current state of the external input pin selected via the secondary count source after application of the IPS bit.

TMR Status/Control Register (SCR) $1FFE80 + $7, $F, $17, $1F	Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Read	TCF	TCFIE	TOF	TOFOE	IEF	IEFIE	IPS	INPUT	CM		MSTR	EEOF	VAL	0	OPS	OEN
	Write														FORCE		
	Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

denotes Reserved Bits

See the following page for continuation of this register

TMR

TMR Status/Control Register (SCR) continued

Bits	Name	Description
7 - 6	CAPTURE MODE	**Input Capture Mode**
		These bits specify the operation of the Capture register and the operation of the input edge flag.
		00 Capture function is disabled
		01 Load Capture register on rising edge of input
		10 Load Capture register on falling edge of input
		11 Load Capture register on any edge of input
5	MSTR	**Master Mode**
		When set, this bit enables Compare function's output to be broadcasted to the other counter/timers in the module. This signal can then be used to reinitialize the other counters and/or force their OFLAG signal outputs.
4	EEOF	**Enable External OFLAG Force**
		When set, this bit enables the Compare from another counter/timer within the same module to force the state of this counter's OFLAG output signal.
3	VAL	**Forced OFLAG Value**
		This bit determines the value of the OFLAG output signal when a software triggered FORCE command, or another counter/timer set as a master, issues a FORCE command.
2	FORCE	**Force OFLAG Output**
		This write-only bit forces the current value of the VAL bit to be written to the OFLAG output. This bit is read as 0. The VAL and FORCE bits can be written simultaneously in a single write operation. Write to the FORCE bit only when the counter is disabled.
		0 No action
		1 Forces the current value of the Val bit to be written to OFLAG output
1	OPS	**Output Polarity Select**
		This bit determines the polarity of the OFLAG output signal.
		0 True polarity
		1 Inverted polarity
0	OEN	**Output Enable**
		When set, this bit enables the OFLAG output signal to be placed on the external pin. Setting this bit connects a timer's output pin to its input. Polarity of the signal will be determined by OPS bit.

TMRA0_SCR (Timer A, Channel 0 Status/Control)—Address: TMRA_BASE + $7
TMRA1_SCR (Timer A, Channel 1 Status/Control)—Address: TMRA_BASE + $F
TMRA2_SCR (Timer A, Channel 2 Status/Control)—Address: TMRA_BASE + $17
TMRA3_SCR (Timer A, Channel 3 Status/Control)—Address: TMRA_BASE + $1F

TMR Status/Control Register (SCR) $1FFE80 + $7, $F, $17, $1F	Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Read	TCF	TCFIE	TOF	TOFOE	IEF	IEFIE	IPS	INPUT	CAPTURE MODE		MSTR	EEOF	VAL	0	OPS	OEN
	Write														FORCE		
	Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

denotes Reserved Bits

Application: _____ Date: _____

_____ Programmer: _____

TOD

TOD Control Status (TODCS)

Bits	Name	Description	
15	**TODSIO**	**Time-of-Day 1-Second Interrupt Occurred Flag**	
		This bit is set when one-second interrupt occurs. This bit is cleared by writing a 0 to the bit position. It must be cleared before exiting the Interrupt Service Routine (ISR).	
14	**TODAL**	**Time-of-Day Alarm Interrupt Occurred Flag**	
		This bit is set when TOD Alarm Interrupt occurs. This bit is cleared by writing a zero to the bit position. This bit must be cleared before exiting the Interrupt Service Routine.	
9 - 8	TEST	TEST Bit Field	
		This bit field is reserved as factory test bits and must be written as 0.	
7	**TODDA**	**Time-of-Day Days Alarm Enable**	
		0	Alarm interrupt ignores days counter
		1	Alarm interrupt requires match of days alarm register to days counter
6	**TODHA**	**Time-of-Day Hours Alarm Enable**	
		0	Alarm interrupt ignores hours counter
		1	Alarm interrupt requires match of hours alarm register to hours counter
5	**TODMA**	**Time-of-Day Minutes Alarm Enable**	
		0	Alarm interrupt ignores minutes counter
		1	Alarm interrupt requires match of minutes alarm register to minutes counter
4	**TODSA**	**Time-of-Day Seconds Alarm Enable**	
		0	Alarm interrupt ignores seconds counter
		1	Alarm interrupt requires match of seconds alarm register to seconds counter
3	**TODSEN**	**Time-of-Day Seconds Interrupt Enable**	
		0	Disables TOD seconds interrupt
		1	Enables TOD seconds interrupt
2	**TODAEN**	**Time-of-Day Alarm Interrupt Enable**	
		0	Disables TOD alarm interrupt
		1	Enables TOD alarm interrupt
1	**TOD_LOCK**	**Time-of-Day Lock**	
		0	TOD registers not locked
		1	TOD registers locked
0	**TODEN**	**Time-of-Day Enable**	
		Writing 0 to TODEN disables the time-of-day counting; the time registers will contain the final value of the time counters and the registers can receive writing.	

TOD Control Status Register (TODCS) $1FFFC0 + $0	Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Read	TOD SIO	TOD AL					TEST		TOD DA	TOD HA	TOD MA	TOD SA	TOD SEN	TOD AEN	TOD_ LOCK	TOD EN
	Write																
	Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

denotes Reserved Bits

TOD

TOD Clock Scaler (TODCSL)

Bits	Name	Description
15 - 0	TODSCL	**Time-of-Day Clock Scaler**
		Setting this field to X divides the TOD Input Clock frequency by X + 1 to produce the time-base clock used for incrementing the counters. The clock generation system and this scaler must be configured so the time-base clock is precisely 1Hz for correct operation of the TOD module as the seconds counter increments with each time-base clock. This register is always readable. It accepts writing only when TODEN = 0.

TOD Clock Scaler Register (TODCSL) $1FFFC0 + $1	Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Read	TIME -OF-DAY CLOCK SCALER															
	Write																
	Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TOD

TOD Seconds Register (TODSEC)

Bits	Name	Description
5 - 0	TODSEC	**Time-of-Day Seconds**
		When TODEN is set, this register is continuously updated to contain the current seconds counter value. The value of the counter can be read through the seconds register, but it cannot be modified. When TODEN is cleared, TOD counting is disabled. This register can then be read or written. When TODEN is set to one again, the counting resumes from the current register value.

TOD Seconds Counter Register (TODSEC) $1FFFC0 + $2	Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Read											TODSEC					
	Write																
	Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

denotes Reserved Bits

TOD

TOD Seconds Alarm Register (TODSAL)

Bits	Name	Description
5 - 0	TODSAL	Time-of-Day Seconds Alarm
		When the value contained in this register matches the value of the seconds counter, the seconds alarm is asserted if the Seconds Alarm Enable (TODSA) bit and Alarm Interrupt Enable (TODAEN) bits are set and all other enabled alarm registers also match.

TOD Seconds Alarm Register (TODSAL) $1FFFC0 + $3	Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Read											TODSAL					
	Write																
	Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

denotes Reserved Bits

TOD

TOD Minutes Register (TODMIN)

Bits	Name	Description
5 - 0	TODMIN	Time-of-Day Minutes
		When TODEN is set, this register is continuously updated to contain the current minutes counter value. The value of the counter can be read through the minutes register, but it cannot be modified. When TODEN is cleared, TOD counting is disabled. This register can then be read or written. When TODEN is set to one again, the counting resumes from the current register value.

TOD Minutes Register (TODMIN) $1FFFC0 + $4	Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Read											TODMIN					
	Write																
	Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

denotes Reserved Bits

TOD

TOD Minutes Alarm Register (TODMAL)

Bits	Name	Description
5 - 0	TODMAL	**Time-of-Day Minutes Alarm**
		When the value contained in this register matches the value of the minutes counter, the minutes alarm is asserted if the Minutes Alarm Enable (TODMA) bit and Alarm Interrupt Enable (TODAEN) bit are both set, and all other enabled alarm registers also match the current time.

TOD Minutes Alarm Register (TODMAL) $1FFFC0 + $5	Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Read											TODMAL					
	Write																
	Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

denotes Reserved Bits

TOD

TOD Hours Register (TODHR)

Bits	Name	Description
5 - 0	TODHR	Time-of-Day Hours
		When TODEN is set, this register is continuously updated to contain the current hours counter value. The value of the counter can be read through the hours register, but it can not be modified. When TODEN is cleared, TOD counting is disabled. This register can then be read or written. When TODEN is set to one again, the counting resumes from the current register value.

TOD Hours Register (TODHR) $1FFFC0 + $6	Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Read											TODHR					
	Write																
	Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

denotes Reserved Bits

TOD

TOD Hours Alarm Register (TODHAL)

Bits	Name	Description
4 - 0	TODHAL	**Time-of-Day Hours Alarm**
		When the value contained in this register matches the value of the hours counter, the hours alarm is asserted if the Hours Alarm Enable (TODHA) bit and the Alarm Interrupt Enable (TODAEN) bit are both set and all other enabled alarm registers also match the current time.

TOD Hours Alarm Register (TODHR) $1FFFC0 + $7	Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Read												TODHAL				
	Write																
	Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

denotes Reserved Bits

TOD

TOD Days Register (TODDAY)

Bits	Name	Description
15 - 0	TODDAY	**Time-of-Day Days**
		When TODEN is set, this counter is continuously updated to contain the current days counter value. The value of the counter can be read through the days register, but it cannot be modified. When TODEN is cleared, TOD counting is disabled. This counter can then be read or written. When TODEN is set to one again, counting will resume counting from the current register value.

TOD Days Register (TODAY) $1FFFC0 + $8	Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Read	TODDAY															
	Write																
	Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TOD

TOD Days Alarm Register (TODDAL)

Bits	Name	Description
15 - 0	TODDAL	Time-of-Day Days
		When TODEN is set, this counter is continuously updated to contain the current days counter value. The value of the counter can be read through the days register, but it cannot be modified. When TODEN is cleared, TOD counting is disabled. This counter can then be read or written. When TODEN is set to one again, counting will resume counting from the current register value.

TOD Days Alarm Register (TODAY) $1FFFC0 + $9	Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Read	TODDAL															
	Write																
	Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPIO

GPIO Port A Peripheral Enable Register (GPIOA_PER)

Bits	Name	Description	
3 - 0	PE	**Port A Peripheral Enable**	
		This bit field controls whether a given pin is in Normal or GPIO modes.	
		0	GPIO mode; pin operation is controlled by GPIO registers
		1	Normal mode; pin operation is controlled by the EMI module

Peripheral Enable Register (GPIOA_PER) $1FFE60 + $0	Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Read	1	1	1	1	1	1	1	1	1	1	1	1	PE			
	Write																
	Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

denotes Reserved Bits

GPIO

GPIO Port B Peripheral Enable Register (GPIOB_PER)

Bits	Name	Description	
15 - 0	**PE**	**Port B Peripheral Enable**	
		This bit field controls whether a given pin is in Normal or GPIO modes	
		0	GPIO mode; pin operation is controlled by GPIO registers
		1	Normal mode; pin operation is controlled by the Host Interface module

Peripheral Enable Register (GPIOB_PER) $1FFE60 + $4	Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Read	PE															
	Write																
	Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

GPIO

GPIO Port C Peripheral Enable Register (GPIOC_PER)

Bits	Name	Description	
5 - 0	PE	**Port C Peripheral Enable**	
		This bit field controls whether a given pin is in Normal or GPIO modes.	
		0	GPIO mode; pin operation is controlled by GPIO registers
		1	Normal mode; pin operation is controlled by the ESSI 0 module

Peripheral Enable Register (GPIOC_PER) $1FFE60 + $8	Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Read	1	1	1	1	1	1	1	1	1	1	PE					
	Write																
	Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

denotes Reserved Bits

GPIO

GPIO Port D Peripheral Enable Register GPIOD_PER)

Bits	Name	Description	
5 - 0	PE	**Port D Peripheral Enable**	
		This bit field controls whether a given pin is in Normal or GPIO modes.	
		0	GPIO mode; pin operation is controlled by GPIO registers
		1	Normal mode; pin operation is controlled by the ESSI 1 module

Peripheral Enable Register (GPIOD_PER) $1FFE60 + $C	Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Read	1	1	1	1	1	1	1	1	1	1	PE					
	Write																
	Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

denotes Reserved Bits

GPIO

GPIO Port E Peripheral Enable Register (GPIOE_PER)

Bits	Name	Description	
3 - 0	PE	**Port E Peripheral Enable**	
		This bit field controls whether a given pin is in Normal or GPIO modes.	
		0	GPIO mode; pin operation is controlled by GPIO registers
		1	Normal mode; pin operation is controlled by the SCI module

Peripheral Enable Register (GPIOE_PER) $1FFE60 + $10	Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Read	1	1	1	1	1	1	1	1	1	1	1	1	PE			
	Write																
	Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

denotes Reserved Bits

GPIO

GPIO Port F Peripheral Enable Register (GPIOF_PER)

Bits	Name	Description	
3 - 0	PE	**Port F Peripheral Enable**	
		This bit field controls whether a given pin is in Normal or GPIO modes.	
		0	GPIO mode; pin operation is controlled by GPIO registers
		1	Normal mode; pin operation is controlled by the SPI module

Peripheral Enable Register (GPIOF_PER) $1FFE60 + $14	Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Read	1	1	1	1	1	1	1	1	1	1	1	1	PE			
	Write																
	Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

denotes Reserved Bits

GPIO

GPIO Port G Peripheral Enable Register (GPIOG_PER)

Bits	Name	Description	
3 - 0	PE	**Port G Peripheral Enable**	
		This bit field controls whether a given pin is in Normal or GPIO modes.	
		0	GPIO mode; pin operation is controlled by GPIO registers
		1	Normal mode; pin operation is controlled by the TMR module

Port G Peripheral Enable Register (GPIOG_PER) $1FFE60 + $18	Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Read	1	1	1	1	1	1	1	1	1	1	1	1	PE			
	Write																
	Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

denotes Reserved Bits

GPIO

GPIO Port H Peripheral Enable Register (GPIOH_PER)

Bits	Name	Description	
2 - 0	PE	**Port H Peripheral Enable**	
		This bit field controls whether a given pin is in Normal or GPIO modes.	
		0	GPIO mode; pin operation is controlled by GPIO registers
		1	Normal mode; pin operation is controlled by the SIM module

Peripheral Enable Register (GPIOH_PER) $1FFE60 + $1C	Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Read	1	1	1	1	1	1	1	1	1	1	1	1	1	PE		
	Write																
	Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

denotes Reserved Bits

GPIO

GPIO Port A Data Direction Register (GPIOA_DDR)

Bits	Name	Description	
3 - 0	DD	**Port A Data Direction**	
		These bits control the pins direction when in GPIO mode. In the Normal mode, these bits have no effect on the output enables or pull-up enables.	
		0	Pin is an input; pull-ups are dependent on value of PUE registers. (default)
		1	Pin is an output; pull-ups are disabled

Data Direction Register (GPIOA_DDR) $1FFE60 + $1	Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Read	0	0	0	0	0	0	0	0	0	0	0	0	DD			
	Write																
	Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

denotes Reserved Bits

GPIO

GPIO Port B Data Direction Register (GPIOB_DDR)

Bits	Name	Description	
15 - 0	DD	**Port B Data Direction**	
		These bits control the pins direction when in GPIO mode. In the Normal mode, these bits have no effect on the output enables or pull-up enables.	
		0	Pin is an input; pull-ups are dependent on value of PUE registers. (default)
		1	Pin is an output; pull-ups are disabled

Data Direction Register (GPIOB_DDR) $1FFE60 + $5	Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Read	DD															
	Write																
	Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPIO

GPIO Port C Data Direction Register GPIOC_DDR)

Bits	Name	Description	
5 - 0	DD	**Port C Data Direction**	
		These bits control the pins direction when in GPIO mode. In the Normal mode, these bits have no effect on the output enables or pull-up enables.	
		0	Pin is an input; pull-ups are dependent on value of PUE registers. (default)
		1	Pin is an output; pull-ups are disabled

Data Direction Register (GPIOC_DDR) $1FFE60 + $9	Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Read	0	0	0	0	0	0	0	0	0	0	DD					
	Write																
	Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

denotes Reserved Bits

GPIO

GPIO Port D Data Direction Register (GPIOD_DDR)

Bits	Name	Description	
5 - 0	DD	**Port D Data Direction**	
		These bits control the pins direction when in GPIO mode. In the Normal mode, these bits have no effect on the output enables or pull-up enables.	
		0	Pin is an input; pull-ups are dependent on value of PUE registers. (default)
		1	Pin is an output; pull-ups are disabled

Data Direction Register (GPIOD_DDR) $1FFE60 + $D	Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Read	0	0	0	0	0	0	0	0	0	0	DD					
	Write																
	Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

denotes Reserved Bits

GPIO

GPIO Port E Data Direction Register (GPIOE_DDR)

Bits	Name	Description	
3 - 0	DD	**Port E Data Direction**	
		These bits control the pins direction when in GPIO mode. In the Normal mode, these bits have no effect on the output enables or pull-up enables.	
		0	Pin is an input; pull-ups are dependent on value of PUE registers. (default)
		1	Pin is an output; pull-ups are disabled

Data Direction Register (GPIOE_DDR) $1FFE60 + $11	Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Read	0	0	0	0	0	0	0	0	0	0	0	0	DD			
	Write																
	Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

denotes Reserved Bits

GPIO

GPIO Port F Data Direction Register (GPIOF_DDR)

Bits	Name	Description	
3 - 0	DD	**Port F Data Direction**	
		These bits control the pins direction when in GPIO mode. In the Normal mode, these bits have no effect on the output enables or pull-up enables.	
		0	Pin is an input; pull-ups are dependent on value of PUE registers. (default)
		1	Pin is an output; pull-ups are disabled

Data Direction Register (GPIOF_DDR) $1FFE60 + $15	Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Read	0	0	0	0	0	0	0	0	0	0	0	0	DD			
	Write																
	Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

denotes Reserved Bits

GPIO

GPIO Port G Data Direction Register (GPIOG_DDR)

Bits	Name	Description	
3 - 0	DD	**Port G Data Direction**	
		These bits control the pins direction when in GPIO mode. In the Normal mode, these bits have no effect on the output enables or pull-up enables.	
		0	Pin is an input; pull-ups are dependent on value of PUE registers. (default)
		1	Pin is an output; pull-ups are disabled

Data Direction Register (GPIOG_DDR) $1FFE60 + $19	Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Read	0	0	0	0	0	0	0	0	0	0	0	0	DD			
	Write																
	Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

denotes Reserved Bits

GPIO

GPIO Port H Data Direction Register (GPIOH_DDR)

Bits	Name	Description	
2 - 0	DD	**Port H Data Direction**	
		These bits control the pins direction when in GPIO mode. In the Normal mode, these bits have no effect on the output enables or pull-up enables.	
		0	Pin is an input; pull-ups are dependent on value of PUE registers. (default)
		1	Pin is an output; pull-ups are disabled

Data Direction Register (GPIOH_DDR) $1FFE60 + $1D	Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Read	0	0	0	0	0	0	0	0	0	0	0	0	0	DD		
	Write																
	Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

denotes Reserved Bits

GPIO

GPIO Port A Data Register (GPIOA_DR)

Bits	Name	Description
3 - 0	DATA	**Port A Data**
		These bits control the output data while in the GPIO mode.

Data Register (GPIOA_DR) $1FFE60 + $2	Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Read	0	0	0	0	0	0	0	0	0	0	0	0	DATA			
	Write																
	Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

denotes Reserved Bits

GPIO

GPIO Port B Data Register (GPIOB_DR)

Bits	Name	Description
15 - 0	DATA	**Port C Data**
		These bits control the output data while in the GPIO mode.

Data Register (GPIOB_DR) $1FFE60 + $6	Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Read	DATA															
	Write																
	Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

denotes Reserved Bits

GPIO

GPIO Port C Data Register (GPIOC_DR)

Bits	Name	Description
5 - 0	DATA	**Port C Data**
		These bits control the output data while in the GPIO mode.

Data Register (GPIOC_DR) $1FFE60 + $A	Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Read	0	0	0	0	0	0	0	0	0	0	DATA					
	Write																
	Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

denotes Reserved Bits

GPIO

GPIO Port D Data Register (GPIOD_DR)

Bits	Name	Description
5 - 0	**DATA**	**Port D Data**
		These bits control the output data while in the GPIO mode.

Data Register (GPIOD_DR) $1FFE60 + $E	Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Read	0	0	0	0	0	0	0	0	0	0	DATA					
	Write																
	Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

denotes Reserved Bits

GPIO

GPIO Port E Data Register (GPIOE_DR)

Bits	Name	Description
3 - 0	DATA	**Port E Data**
		These bits control the output data while in the GPIO mode.

Data Register (GPIOE_DR) $1FFE60 + $12	Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Read	0	0	0	0	0	0	0	0	0	0	0	0	DATA			
	Write																
	Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

denotes Reserved Bits

GPIO

GPIO Port F Data Register (GPIOF_DR)

Bits	Name	Description
3 - 0	DATA	**Port F Data**
		These bits control the output data while in the GPIO mode.

Data Register (GPIOF_DR) $1FFE60 + $16	Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Read	0	0	0	0	0	0	0	0	0	0	0	0	DATA			
	Write																
	Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

denotes Reserved Bits

GPIO

GPIO Port G Data Register (GPIOG_DR)

Bits	Name	Description
3 - 0	DATA	**Port G Data**
		These bits control the output data while in the GPIO mode.

Data Register (GPIOG_DR) $1FFE60 + $1A	Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Read	0	0	0	0	0	0	0	0	0	0	0	0	DATA			
	Write																
	Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

denotes Reserved Bits

GPIO

GPIO Port H Data Register (GPIOH_DR)

Bits	Name	Description
3 - 0	DATA	**Port H Data**
		These bits control the output data while in the GPIO mode.

Data Register (GPIOH_DR) $1FFE60 + $1E	Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Read	0	0	0	0	0	0	0	0	0	0	0	0	DATA			
	Write																
	Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

denotes Reserved Bits

GPIO

GPIO Port A Pull-Up Enable Register (GPIOA_PUE)

Bits	Name	Description	
3 - 0	PUE	**Port A Pull-Up Enable**	
		These bits control whether pull ups are enabled for inputs in either Normal or GPIO mode.	
		0	Pull ups disabled for inputs
		1	Pull ups enabled for inputs (default)

Pull-Up Enable Register (GPIOA_PUE) $1FFE60 + $3	Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Read	1	1	1	1	1	1	1	1	1	1	1	1	PUE			
	Write																
	Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

denotes Reserved Bits

GPIO

GPIO Port B Pull-Up Enable Register (GPIOB_PUE)

Bits	Name	Description	
15 - 0	**PUE**	**Port B Pull-up Enable**	
		These bits control whether pull ups are enabled for inputs in either Normal or GPIO mode.	
		0	Pull ups disabled for inputs
		1	Pull ups enabled for inputs (default)

Pull-Up Enable Register (GPIOB_PUE) $1FFE60 + $7	Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Read	PUE															
	Write																
	Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

denotes Reserved Bits

GPIO

GPIO Port C Pull-Up Enable Register (GPIOC_PUE)

Bits	Name	Description	
5 - 0	PUE	**Port C Pull-up Enable**	
		These bits control whether pull ups are enabled for inputs in either Normal or GPIO mode.	
		0	Pull ups disabled for inputs
		1	Pull ups enabled for inputs (default)

Pull-Up Enable Register (GPIOC_PUE) $1FFE60 + $B	Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Read	1	1	1	1	1	1	1	1	1	1	PUE					
	Write																
	Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

denotes Reserved Bits

GPIO

GPIO Port D Data Register GPIOD_PUE)

Bits	Name	Description	
5 - 0	PUE	**Port D Pull-Up Enable**	
		These bits control whether pull ups are enabled for inputs in the Normal or GPIO modes.	
		0	Pull ups disabled for inputs
		1	Pull ups enabled for inputs (default)

Pull-Up Enable Register (GPIOD_PUE) $1FFE60 + $F	Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Read	1	1	1	1	1	1	1	1	1	1	PUE					
	Write																
	Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

denotes Reserved Bits

GPIO

GPIO Port E Pull-Up Register (GPIOE_PUE)

Bits	Name	Description	
3 - 0	PUE	**Port E Pull-Up Enable**	
		These bits control whether pull ups are enabled for inputs in the Normal or GPIO modes.	
		0	Pull ups disabled for inputs
		1	Pull ups enabled for inputs (default)

Pull-Up Enable Register (GPIOE_PUE) $1FFE60 + $13	Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Read	1	1	1	1	1	1	1	1	1	1	1	1	PUE			
	Write																
	Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

denotes Reserved Bits

GPIO

GPIO Port F Pull-Up Register (GPIOF_PUE)

Bits	Name	Description	
3 - 0	PUE	**Port F Pull-Up Enable**	
		These bits control whether pull ups are enabled for inputs in the Normal or GPIO modes.	
		0	Pull ups disabled for inputs
		1	Pull ups enabled for inputs (default)

Pull-Up Enable Register (GPIOF_PUE) $1FFE60 + $B	Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Read	1	1	1	1	1	1	1	1	1	1	1	1	PUE			
	Write																
	Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

denotes Reserved Bits

GPIO

GPIO Port G Pull-Up Register (GPIOG_PUE)

Bits	Name	Description	
3 - 0	PUE	**Port G Pull-Up Enable**	
		These bits control whether pull ups are enabled for inputs in the Normal or GPIO modes.	
		0	Pull ups disabled for inputs
		1	Pull ups enabled for inputs (default)

Pull-Up Enable Register (GPIOG_PUE) $1FFE60 + $1B	Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Read	1	1	1	1	1	1	1	1	1	1	1	1	PUE			
	Write																
	Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

denotes Reserved Bits

GPIO

GPIO Port H Pull-Up Register (GPIOH_PUE)

Bits	Name	Description
2 - 0	PUE	**Port H Pull-Up Enable**
		This register is not used because pull ups are always disabled for the MODA, B, and C

Pull-Up Enable Register (GPIOH_PUE) $1FFE60 + $1F	Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Read	1	1	1	1	1	1	1	1	1	1	1	1	1	PUE		
	Write																
	Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

denotes Reserved Bits

HI8

HI8 Host Control Register (HCR)

Bits	Name	Description
8	HRMS	**Host Request Mode Select**
		The Host mode Select (HRMS) bit controls the host request pins
7	HDDS	**Host Dual Data Strobe**
		When the Host Dual Data Strobe (HDDS) bit is set, the HI8 operates in the Dual Data Strobe Bus mode; a host bus with separated read and write data strobes. When cleared, the HI8 operates in the Single Strobe Bus mode, i.e. a Host bus with a single Data Strobe signal.
6	TDMAEN	**DSC Side Transmit DMA Enable**
		The TDMAEN bit is used to enable DSC side Transmit DMA operations. When set, the on-chip DMA controller handles transferring data between the HTX register and DSC memory. The on-chip DMA controller must be appropriately configured to implement the desired data transfer.
5	RDMAEN	**DSC Side Receive DMA Enable**
		The RDMAEN bit is used to enable DSC side Receive DMA operations. When set, the on-chip DMA controller handles transferring data between the HRX register and DSC memory. The on-chip DMA controller must be appropriately configured to implement the desired data transfer.
4	HF2	**Host Flags 2 and 3**
3	HF3	The Host Flag 2 and Host Flag 3 (HF2 and HF3) bits are used as general purpose flags for DSC-to-Host communication. HF2 and HF3 may be set or cleared by the core. HF2 and HF3 are reflected in the Interrupt Status Register (ISR) on the Host Side if they are modified by the software, the host processor can read the modified values by reading the ISR.
2	HCIE	**Host Command Interrupt Enable**
		The HCIE bit is used to enable a core interrupt when the HCP status bit in the HSR is set. When cleared, HCP interrupts are disabled. When set, a Host Command Interrupt request occurs if HCP is set. The interrupt address is determined by the Host Command Vector Register. Cleared on hardware reset.
1	HTIE	**Host Transmit Interrupt Enable**
		The HTIE bit is used to enable a core interrupt when the Host Transmit Data Empty status bit in the HSR is set. When cleared, HTDE interrupts are disabled. When set, a Host Transmit Data Interrupt request occurs when the HTDE bit is set. It is cleared on hardware reset.
0	HRIE	**Host Receive Interrupt Enable**
		The HRIE bit is used to enable a core interrupt when the Host Receive Data Full (HRDF) status bit in the Host Status Register (HSR) is set. When cleared, HRDF interrupts are disabled. When set, a Host Receive Data Interrupt request occurs if the HRDF bit is also set. It is cleared on hardware reset.

HI8 Host Control Register (HCR) $1FFFD8 + $0	Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Read	0	0	0	0	0	0	0	HRMS	HDDS	TDMAEN	RDMAEN	HF3	HF2	HCIE	HTIE	HRIE
	Write	0	0	0	0	0	0	0									
	Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

denotes Reserved Bits

HI8

HI8 Status Register (HSR)

Bits	Name	Description
5	HDMA	**Host DMA Status**
		The Host DMA Status (HDMA) bit indicates the host processor has enabled the Host DMA mode of the HI8 by setting HM1 or HM0 to 1. When the HDMA status bit is set at 0, it indicates the Host DMA mode is disabled by the Host mode bits HM0 and HM1, both having been cleared, in the Interface Control Register ICR and no Host DMA operations are pending. When the HDMA status bit is set, the Host DMA mode is enabled by the Host mode bits HM0 and HM1. The transmit or receive channel not in use can be used by the Host for polled or interrupt operation by the device. HDMA is cleared by a DSC reset.
4	HF1	**Host Flags 0 and 1**
3	HF0	The Host Flag 0-1 (HF0 and HF1) bits are used as a general purpose flags for Host-to-DSC communication. The HF0 and HF1 bits can be set or cleared by the Host. These bits reflect the status of Host Flags HF0 and HF1 in the Interface Control Register (ICR) on the Host Side.
2	HCP	**Host Command Pending**
		The Host Command Pending (HCP) flag bit reflects the status of the HC bit in the Command Vector Register (CVR), indicating a Host Command Interrupt is pending. The HCP bit is set when the HC bit is set, and both bits are cleared by the HI8 hardware when the interrupt request is serviced by the core. The Host can also clear the HC bit, thereby clearing the HCP bit as well. The HCP bit is cleared on hardware reset.
1	HTDE	**Host Transmit Data Empty**
		The Host Transmit Data Empty (HTDE) flag bit indicates the Host Transmit Data (HTX) register is empty and can be written by the core. The HTDE bit is set when the HTX register is transferred to the RXH/RXL registers, and cleared when Host Transfer Date (HTX) is written by the core. When the HTDE bit is set, the HI8 generates a Transmit Data Full DMA request. HTDE can also be set by the host processor using the initialize function. The HTDE bit is set on hardware reset.
0	HRDF	**Host Receive Data Full**
		The Host Receive Data Full (HRDF) flag bit indicates the Host Receive Data (HRX) register contains data from the host processor. The HRDF bit is set when data is transferred from the TXH/TXL registers to the Host Receive Data (HRX) register. The HRDF bit is cleared when the HRX register is read by the core. When the HRDF bit is set, the HI8 generates a receive data full DMA request. The HRDF bit can also be cleared by the host processor using the initialize function. The HRDF bit is cleared on hardware reset.

HI8 Host Status Register (HSR) $1FFFD8 + $1	Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Read	0	0	0	0	0	0	0	0	0	0	HDMA	HF1	HF0	HCP	HTDE	HRDF
	Write	0	0	0	0	0	0	0	0	0	0						
	Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

denotes Reserved Bits

HI8

HI8 Transmit Data Register (HTX)

Bits	Name	Description
15 - 0	HTX	**HI8 Transmit Data Register**
		The write-only HI8 Transmit Data (HTX) register is used for DSC-to-Host data transfers. Writing to the HTX register clears the HTDE bit in the HSR. The device can program the HTIE bit causing a Host Transmit Data interrupt when the HTDE bit is set. The HTX register is transferred as 16-bit data to the receive byte registers RXH/RXL when both the HTDE bit on the DSC Side and the RXDF status bits on the Host Side are cleared. This transfer operation sets both RXDF and HTDE bits. Data should not be written to the HTX register until the HTDE bit is set to prevent the previous data from being overwritten.

HI8 Transmit Data Register (HTX) $1FFFD8 + $2	Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Read																
	Write	HIGH BYTE (FROM HRX)								LOW BYTE (FROM LRX)							
	Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

denotes Reserved Bits

HI8

HI8 Receive Data Register (HRX)

Bits	Name	Description
15 - 0	HRX	**HI8 Receive Data Register**
		The read-only HI8 Receive Data (HRX) register is used for Host-to-DSC data transfers. The HRX register is loaded with 16-bit data from the transmit data registers TXH/TXL on the Host Side when both the transmit data register empty TXDE on the Host Side, and DSC Host Receive Data Full (HRDF) bits are cleared. This transfer operation sets TXDE and HRDF bits. The HRX register contains valid data when the HRDF bit is set. Reading HRX clears HRDF. The DSC may program the HRIE bit to cause a Host Receive Data interrupt when HRDF is set.

HI8 Receive Data Register (HRX) $1FFFD8 + $2	Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Read	HIGH BYTE (FROM HTX)								LOW BYTE (FROM LTX)							
	Write																
	Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

denotes Reserved Bits

HI8

HI8 Interface Control Register (ICR)

Bits	Name	Description
7	INIT	**Initialize**
		The Initialize bit is used by the host processor to force initialization of the HI8 hardware.
6	HM1	**Host Mode Control**
5	HM0	The Host mode control bits HM0 and HM1 select the Transfer mode of the HI8. HM1 and HM0 enable the DMA mode of operation, or they interrupt a no-Host DMA mode of operation when HREQ bit in the Host Control Register (HCR) is set.
4	HF1	**Host Flag 1**
		The Host Flag 1 (HF1) bit is used as a general purpose flag for Host-to-DSC communication. The HF1 bit can be set or cleared by the host processor, but cannot be changed by the core. The HF1 bit is cleared on DSC reset.
3	HF0	**Host Flag 0**
		The Host Flag 0 (HF0) bit is used as a general purpose flag for Host-to-DSC communication. The HF0 bit can be set or cleared by the host processor, but cannot be changed by the core. The HF0 bit is cleared on DSC reset.
2	HLEND	**ICR Host Little Endian**
		This bit allows the HI8 to be accessed by the Host in or Big Endian data order. When the HLEND bit is set, the HI8 can be accessed by the Host in Little Endian order. The RXH/TXH is located at address $7 and RXL/TXL at $6. When the HLEND bit is cleared, the HI8 can be accessed by the Host in Big Endian Host data order. The RXH/TXH is located at address $6 and RXL/TXL at $7. The HLEND bit is cleared on hardware reset.
1	TREQ	**ICR Transmit Request Enable**
		This bit is used to control the HREQ pin for Host transmit data transfers. In the Interrupt mode, and the DMA is off, TREQ is used to enable interrupt requests via the external Host Request (HREQ or HTRQ) pin when the TXDE status bit in the ISR is set. When TREQ is cleared, TXDE interrupt is disabled.
0	RREQ	**Receive Request Enable**
		This bit is used to control the HREQ pin for Host receive data transfers. In the Interrupt mode (HDMA off), the RREQ is used to enable interrupt requests via the external Host Request (HREQ or HRRQ) pin when the Receive Data Register Full (RXDF) status bit in the Interrupt Status register (ISR) is set. When RREQ is cleared, RXDF interrupts are disabled. When RREQ is set, the external Host Request HREQ pin or HRRQ is asserted if RXDF is set in interrupt mode.

HI8 Interface Control Register (ICR) $1FFFD8 + $0	Bits	7	6	5	4	3	2	1	0
	Read	INIT	HM1	HM0	HF1	HF0	HLEND	TREQ	RREQ
	Write								
	Reset	0	0	0	0	0	0	0	0

HI8

HI8 Command Vector Register (CVR)

Bits	Name	Description
7	HC	**Host Command**
		This bit is used by the host processor to handshake the execution of Host Command interrupts. Normally, the host processor sets HC = 1 to request the Host Command interrupt from the core. Setting the HC bit causes Host Command pending to be set in the Host Status Register.
6 - 0	HV	**CVR Host Vector**
		These seven bits select the Host Command interrupt offset address in the vector table to be used by the Host Command interrupt logic. When the Host Command interrupt is recognized by the DSC interrupt control logic the offset address of the interrupt vector table taken is 2 × HVI0, 6. The Host can write HC and HV in the same write cycle.

HI8 Command Vector Register (CVR) $1FFFD8 + $1	Bits	7	6	5	4	3	2	1	0
	Read	HC	HV6	HV5	HV4	HV3	HV2	HV1	HV0
	Write								
	Reset	0	0	0	0	0	0	0	0

HI8

HI8 Interface Status Register (ISR)

Bits	Name	Description
7	HREQ	**Host Request**
		This bit indicates the status of the external Host Request (HREQ) output pin if the HRMS bit is cleared; or the external Host Transmit Receive Request (HTRQ) output pins, and HRRQ respectively, if HRMS is set. When the HREQ status bit is cleared, it indicates the Host Request pin, HREQ or HTRR, HTRQ and HRRQ, are deasserted and either host processor interrupts or host DMA transfers are being requested.
6	DMA	**Host DMA Status**
		The DMA status bit (DMA) indicates that the host processor has enabled the DMA mode of the HI8 (HM1 or HM0 =1). When the DMA status bit is clear, it indicates that the DMA mode is disabled by the Host mode bits (HM0 and HM1) in the Interface Control Register ICR and no DMA operations are pending. When DMA is set, it indicates that the DMA mode is enabled and the host processor should not use the active DMA channel (RXH:RXL or TXH:TXL depending on DMA direction) to avoid conflicts with the DMA data transfers.
4	HF3	**Host Flag 3**
		The Host Flag 3 (HF3) bit in the Interrupt Status Register indicates the state of Host Flag 3 in the Host Control Register on the DSC side. The HF3 bit can only be changed by the DSC Side.
3	HF2	**Host Flag 2**
		The Host Flag 2 (HF2) bit in the Interface Control Register (ISR) indicates the state of Host Flag 2 in the Host Control Register on the DSC side. The HF2 bit can only be changed by the DSC Side.
2	TRDY	**Transmitter Ready**
		This flag bit indicates TXH, TXL, and the HRX registers are empty. When the TRDY bit is set, the data the host processor writes to the TXH and TXL registers is immediately transferred to the DSC side of the HI8. The many applications can use this feature.
1	TXDE	**Transmit Data Register Empty**
		This bit indicates the transmit byte registers (TXH, and TXL) are empty and can be written by the host processor. TXDE is set when the transmit byte registers are transferred to the HRX register. TXDE is cleared when the transmit (TXL or TXH according to HLEND bit) register is written by the host processor. TXDE can be set by the host processor using the initialize feature.
0	RXDF	**Receive Data Register Full**
		This flag bit indicates the receive byte registers (RXH and RXL) contain data from the DSC Side and can be read by the host processor. The RXDF bit is set when the HTX is transferred to the receive byte registers. RXDF is cleared when the receive data (RXL or RXH according to HLEND bit) register is read by the host processor.

HI8 Interface Status Register (ISR) $1FFFD8 + $2	Bits	7	6	5	4	3	2	1	0
	Read	HREQ	DMA	0	HF3	HF2	TRDY	TXDE	RXDF
	Write			0					
	Reset	0	0	0	0	0	0	1	0

denotes Reserved Bits

HI8

HI8 Interrupt Vector Register (IVR)

Bits	Name	Description
7 - 0	**IV7 - IV0**	**Interrupt Vector**
		This read/write register typically containing the interrupt vector number used with MC68000 family processor vectored interrupts. Only the host processor can read and write this register. The contents of IVR are placed on the Host data bus (H0–H7) when both HREQ and HACK pins are asserted and Host DMA is not enabled. The contents of this register are initialized to a pre-defined value by a hardware or software reset, corresponding to the uninitialized interrupt vector in the MC68000 family.

HI8 Interrupt Vector Register (IVR) $1FFFD8 + $3	Bits	7	6	5	4	3	2	1	0
	Read	IV7	IV6	IV5	IV4	IV3	IV2	IV1	IV0
	Write								
	Reset	0	0	0	0	1	1	1	1

HI8

HI8 Receive Byte Registers (RXH, RXL)

Bits	Name	Description
7 - 0	RXH	**Receive Data High**
		This *read-only* register receives data from the High Byte of the HTX register. It is selected by three external Host Address (HA2, HA1, and HA0) inputs during a Host Processor read operation or by an on-chip address counter in DMA operations. The Receive Byte Registers contain valid data when the Receive Data Register Full (RXDF) bit is set. The Host Processor can program the RREQ bit to assert the external HREQ pin when the RXDF bit is set. This informs the Host Processor or Host DMA controller of the Receive Byte Registers full condition. Reading the Data Register at Host Address $7 clears the RXDF bit.

HI8 Receive Byte Registers (RXH) $1FFFD8 + $6/$7*	Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Read									RXH							
	Write																
	Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description
7 - 0	RXL	**Receive Data Low**
		This *read-only* register receives data from the Low Byte of the HTX register. It is selected by three external Host Address (HA2, HA1, and HA0) inputs during a Host Processor read operation or by an on-chip address counter in DMA operations. The Receive Byte Registers contain valid data when the Receive Data Register Full (RXDF) bit is set. The host processor may program the RREQ bit to assert the external HREQ pin when the RXDF bit is set. This informs the Host Processor or Host DMA controller of the Receive Byte Registers full condition. Reading the data register at Host Address $7 clears the RXDF bit.

HI8 Receive Byte Registers (RXL) Base + $7/$6*	Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Read									RXL							
	Write																
	Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

denotes Reserved Bits * Depends on HLEND setting

HI8

HI8 Transmit Byte Registers (TXH, TXL)

Bits	Name	Description
7 - 0	TXH	**Transmit Data High**
		This *write-only* register transmits data from the High Byte of the HRX register. It is selected by three external Host Address (HA2, HA1, and HA0) inputs during a Host Processor write operation or by an on-chip address counter in DMA operations. Data can be written into the Transmit Data Registers when the TXDE bit is set. The Host Processor can program the TREQ bit to assert the external HREQ pin when TXDE bit is set. This informs the Host Processor or Host DMA controller the Transmit Byte Registers are empty. Writing the data register at Host Address $7 clears the TXDE bit.

HI8 Transmit Byte Registers (TXH) Base + $6/$7*	Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Read																
	Write									TXH							
	Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description
7 - 0	TXL	**Transmit Data Low**
		This *write-only* register transmits data from the Low Byte of the HRX register. It is selected by three external Host Address (HA2, HA1, and HA0) inputs during a Host Processor write operation or by an on-chip address counter in DMA operations. Data can be written into the Transmit Data Registers when the TXDE bit is set. The Host Processor can program the TREQ bit to assert the external HREQ pin when TXDE bit is set. This informs the Host Processor or Host DMA controller the Transmit Byte Registers are empty. Writing the data register at Host Address $7 clears the TXDE bit.

HI8 Transmit Byte Registers (TXL) Base + $7/$6*	Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Read																
	Write									TXL							
	Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

denotes Reserved Bits * Depends on HLEND setting

INDEX

Symbols

Numerics

A

B

C

D

E

F

G

H

I

J

L

M

N

O

P

Q

R

S

T

U

V

W

X

Y

Z

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
1-800-441-2447 or 303-675-2140
Fax: 303-675-2150
LDCForFreescaleSemiconductor@hibbertgroup.com

DSP5685XUM
Rev. 4
7/2005