Order this document
by BR756/D

MOTOROLA
SEMICONDUCTOR
TECHNICAL DATA

MC68332

Technical Summary
32-Bit Microcontroller

The MC68332, a 32-bit highly integrated microcontroller, combines high-performance data manipulation capabilities with powerful peripheral subsystems: the system integration module (SIM), the queued serial module (QSM), and the time processor unit (TPU). All are connected to the CPU32 through the intermodule bus (IMB). The MC68332 is the first member of the M68300 Family of modular embedded controllers featuring fully static, high-speed complementary metal-oxide semiconductor (CMOS) technology. Based on the powerful MC68020, the CPU32 instruction processing module provides enhanced system performance and also uses the extensive software base for the Motorola M68000 Family.

Features

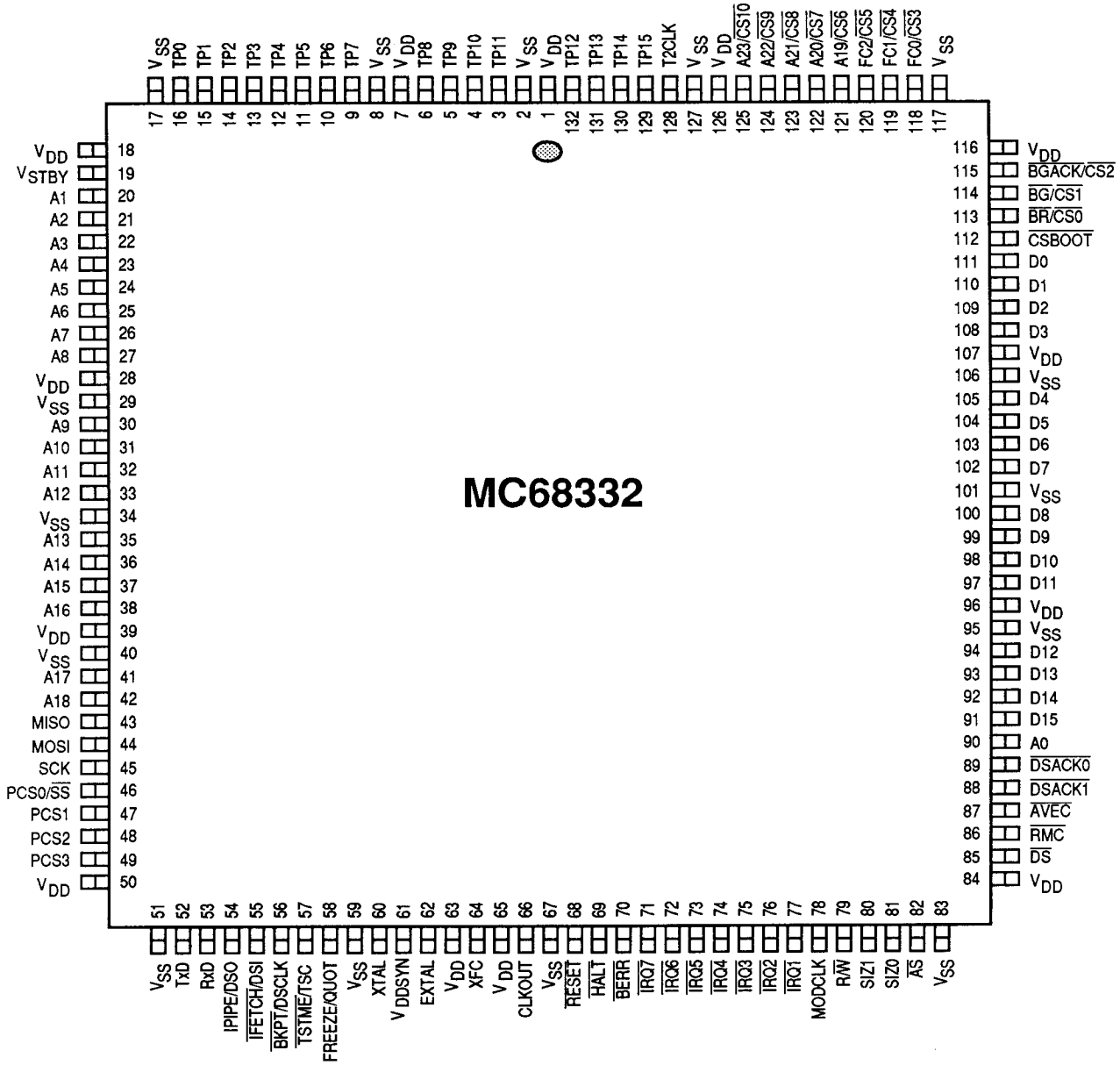
- Modular Architecture in a Single Chip
- CPU: 32-Bit M68000 Family (CPU32):
  - — Upward Object-Code Compatible from the MC68010
  - — New Instruction for Controller Applications
- Low-Power Operation:
  - — 600 mW Maximum
  - — 500 µW in Standby Mode
- Frequency, Software Programmable:
  - — On-Chip Phase-Locked Loop (PLL), 131 kHz to 16.78 MHz (5-V Supply)
  - —Uses Inexpensive 32.768-kHz Watch Crystal
- Technology:
  - — 1-Micron High-Density Complementary Metal-Oxide Semiconductor (HCMOS)
  - — Static Design
- Transistor Count: 422,000
- Package:
  - — 132-Pin Plastic Quad Flat Pack (PQFP)
- Intelligent 16-Bit Time Processor Unit (TPU):
  - — Dedicated Microengine Operating Independently of CPU32
  - — 16 Independent, Programmable Channels and Pins
  - — Any Channel Can Perform Any Time Function (Input Capture (IC), Output Compare (OC), Pulse-Width Modulation (PWM), Stepper Motor (SM), etc.)
  - — Two Timer Count Registers with Programmable Prescalers
  - — Selectable Channel Priority Levels
- Two Serial Input/Output (I/O) Subsystems:
  - — Enhanced Serial Communications Interface (SCI) Universal Asynchronous Receiver Transmitter (UART) with Parity and Programmable Baud Rate Modulus Counter
  - — Enhanced Serial Peripheral Interface with I/O Queue (QSPI)
- On-Chip Memory: 2K Bytes Standby RAM
- On-Chip, Programmable, Chip-Select Logic:
  - — Up to 12 Signals for Memory and Peripheral I/O
- System Failure Protection:
  - — Software Watchdog Timer
  - — Periodic Interrupt Timer
  - — M68000 Family Spurious Interrupt, HALT, and Bus Timeout Monitors
- Up to 32 Discrete I/O Pins

This document contains information on a new product. Specifications and information herein are subject to change without notice.

MOTOROLA

This Page Intentionally Left Blank

Table of Contents

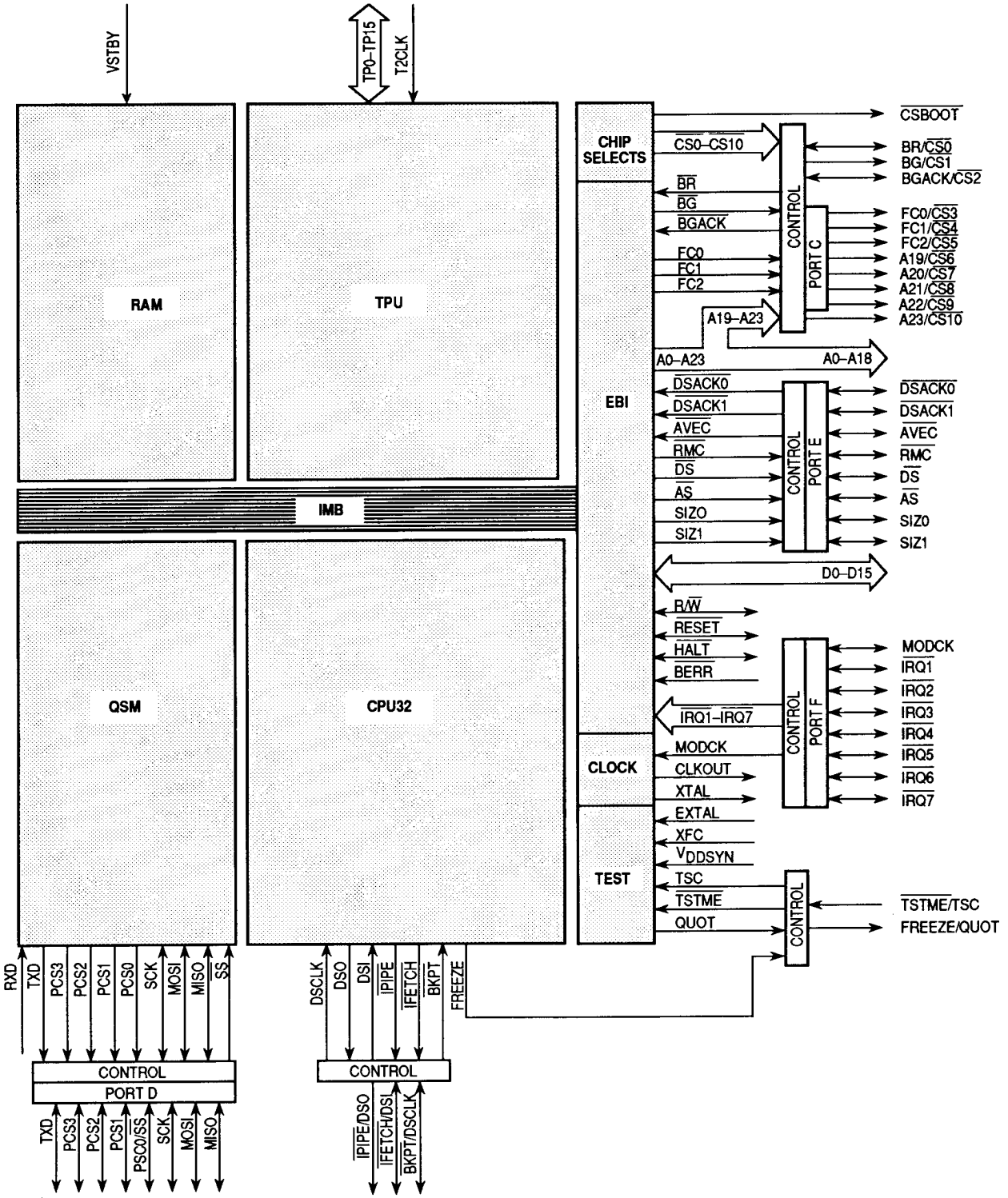


Pin Assignments

Standard MC68332 Ordering Information

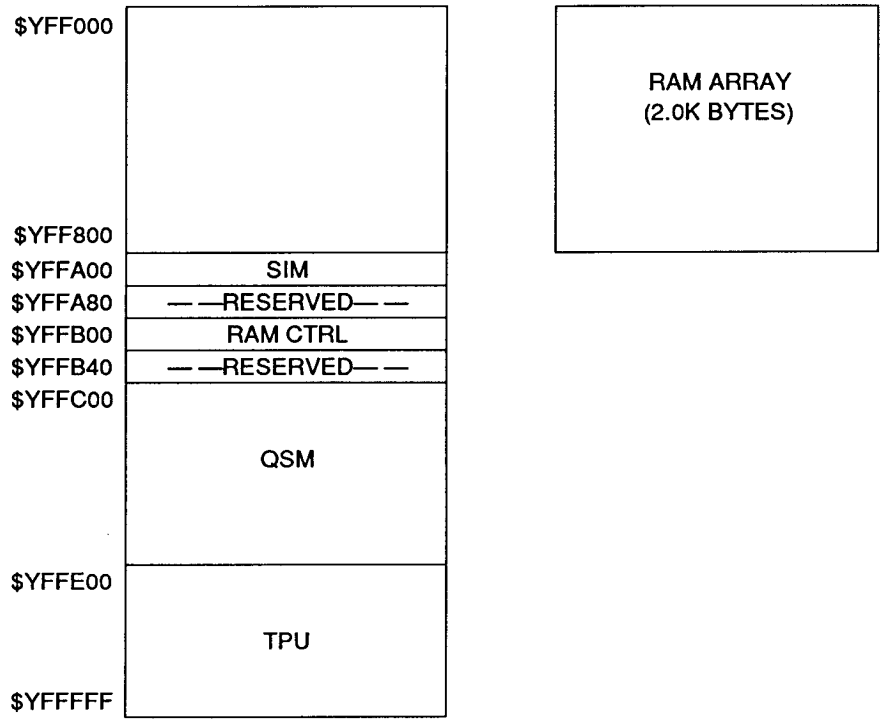
Package Type	Frequency (MHz)	Temperature	Order Number
Ceramic Surface Mount FE Suffix	16.78	–40°C to +85°C	XC68332FE (now) MC68332FE (1Q91) SPAKXC68332FE* SPAKMC68332FE*
Plastic Surface Mount FC Suffix	16.78	–40°C to +85°C	XC68332FC (1Q91) MC68332FC (2Q91) SPAKXC68332FC* SPAKMC68332FC*

*These are sample packs and must be ordered in multiples of two parts and are shipped in special cases.



MC68332 Block Diagram

MC68332 Module Memory Map



Module	Size (Bytes)	Address Bus Decoding						Base Address
		A23	—	A12	A11	—	A0	
SIM	128	M111	1111	1111	1010	0XXX	XXXX	$YFFA00
RAM CTRL	64	M111	1111	1111	1011	00XX	XXXX	$YFFB00
QSM	512	M111	1111	1111	110X	0XXX	XXXX	$YFFC00
TPU	512	M111	1111	1111	111X	0XXX	XXXX	$YFFE00

Y = M111, where M is the modmap signal state on the IMB, which reflects the state of the modmap bit in the module configuration register of the system integration module (Y = $7 or $F).

Central Processor Unit (CPU32)

The CPU32 is fully object code compatible with the M68000 Family, which excels at processing calculation-intensive algorithms and supporting high-level languages.The CPU32 supports all of the MC68010 and most of the MC68020 enhancements, such as virtual memory support, loop mode operation, instruction pipeline, and 32-bit mathematical operations. Powerful addressing modes provide compatibility with existing software programs and increase the efficiency of high-level language compilers. New instructions, such as table lookup and interpolate, and low-power stop, support the specific requirements of controller applications. Also included is the background debug mode, an alternate operating mode that suspends normal operation and allows the CPU to accept debugging commands from the development system.

Features

- Fully Object Code Compatible with M68000 Family
- Virtual Memory Implementation (Instruction Restart Method)
- Loop Mode of Instruction Execution (Faster Execution of Certain Instructions)
- Fast Multiply, Divide, and Shift Instructions
- Fast Bus Interface with Dynamic Bus Port Sizing
- Improved Exception Handling for Controller Applications
- Enhanced Addressing Modes
  - — Scaled Index
  - — Address Register Indirect with Base Displacement and Index
  - — Expanded PC Relative Modes
  - — 32-Bit Branch Displacements
- Instruction Set Enhancements
  - — High-Precision Multiply and Divide
  - — Trap on Condition Codes
  - — Upper and Lower Bounds Checking
  - — Enhanced Breakpoint Instruction
- Table Lookup and Interpolate Instruction
- Low-Power Stop Instruction
- Hardware Breakpoint Signal, Background Mode
- 16.78-MHz Operating Frequency at –40 to 85°C
- Fully Static Implementation

Architecture Summary

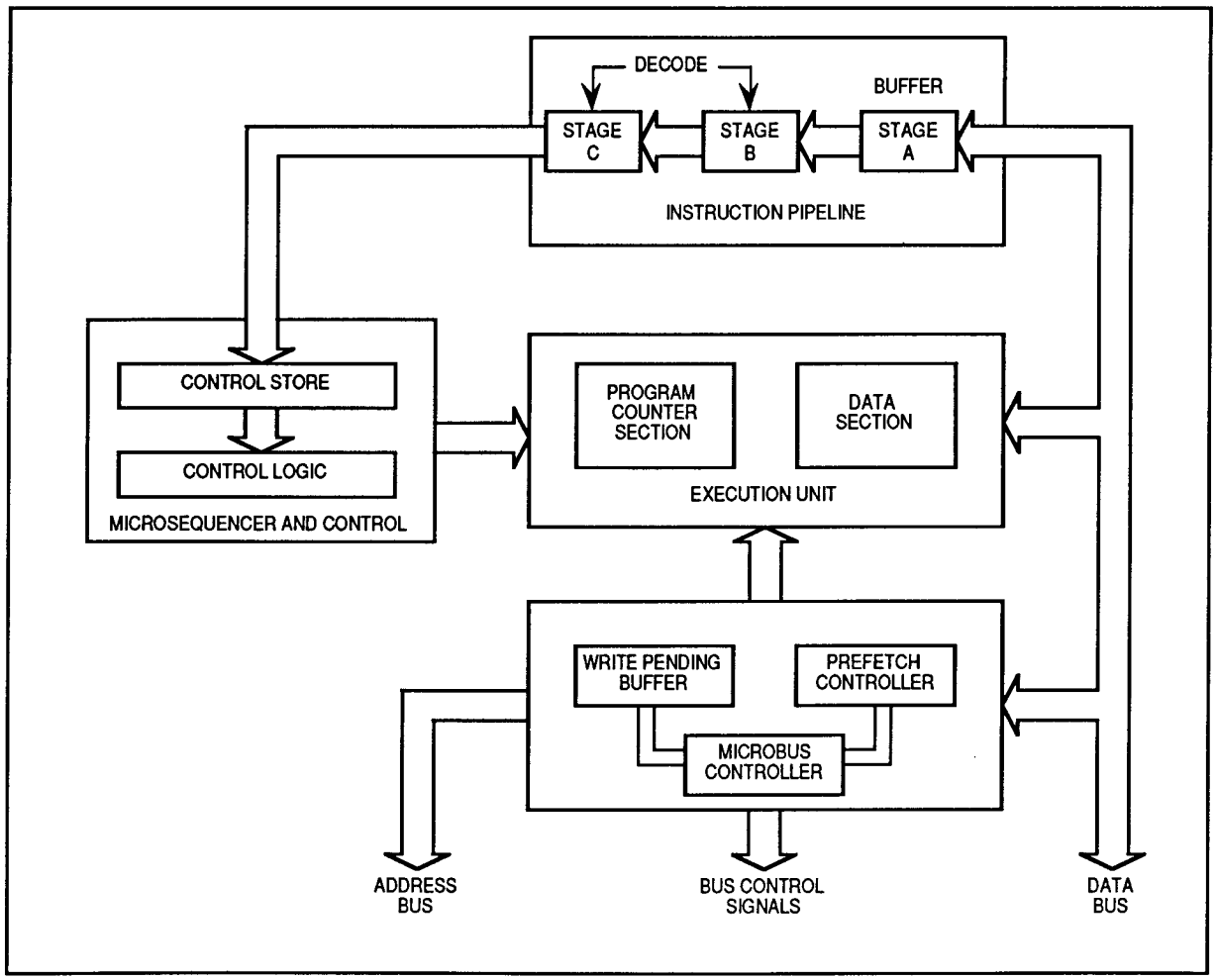
The CPU32 architecture includes several important features that provide both power and versatility to the user. The CPU32 is source and object code compatible with the MC68000 and MC68010. All user state programs can be executed unchanged. The major CPU32 features are as follows:

- 32-Bit Internal Data Path and Arithmetic Hardware — 16-Bit External Data Bus
- 32-Bit Internal Address Bus — 24-Bit External Address Bus
- Rich Instruction Set
- Eight 32-Bit General-Purpose Data Registers
- Seven 32-Bit General-Purpose Address Registers
- Separate User and Supervisor Stack Pointers and Address Spaces
- Separate Program and Data Address Spaces
- Flexible Addressing Modes
- Full Interrupt Processing

Data Types

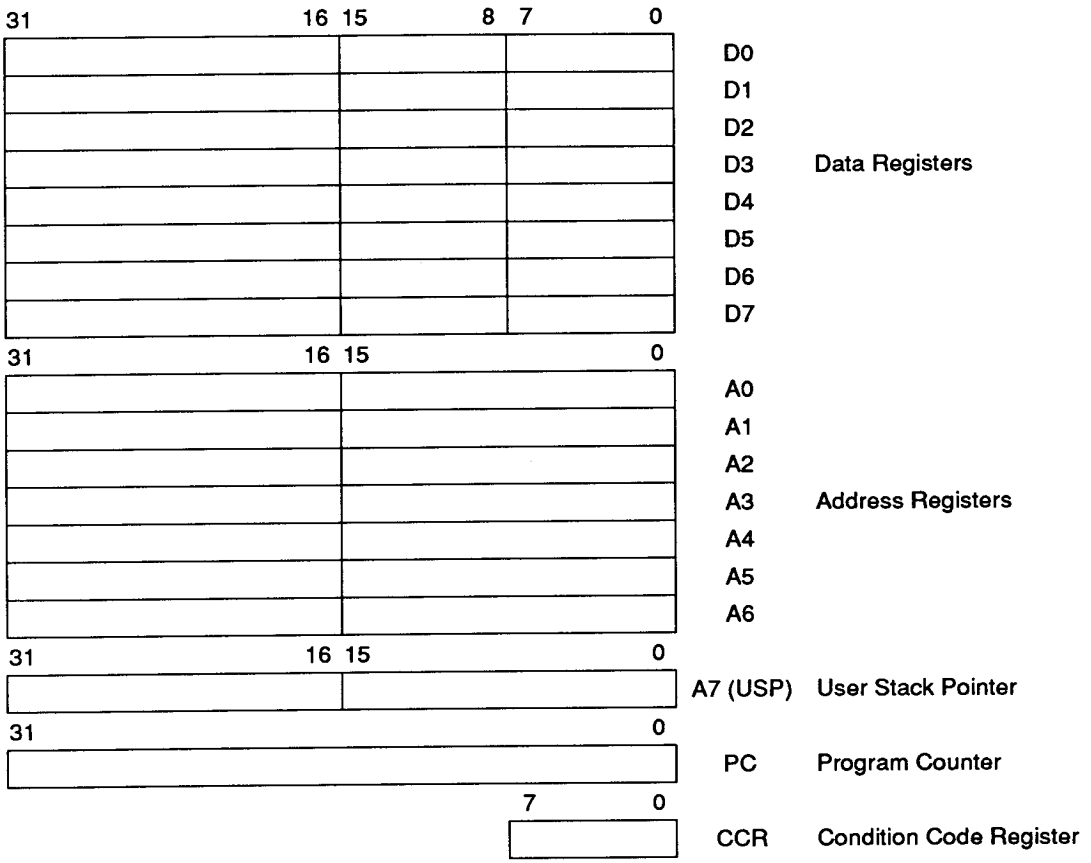
Six basic data types are supported:

- Bits
- Packed Binary Coded Decimal Digits
- Byte Integers (8 bits)
- Word Integers (16 bits)
- Long-Word Integers (32 bits)
- Quad-Word Integers (64 bits)

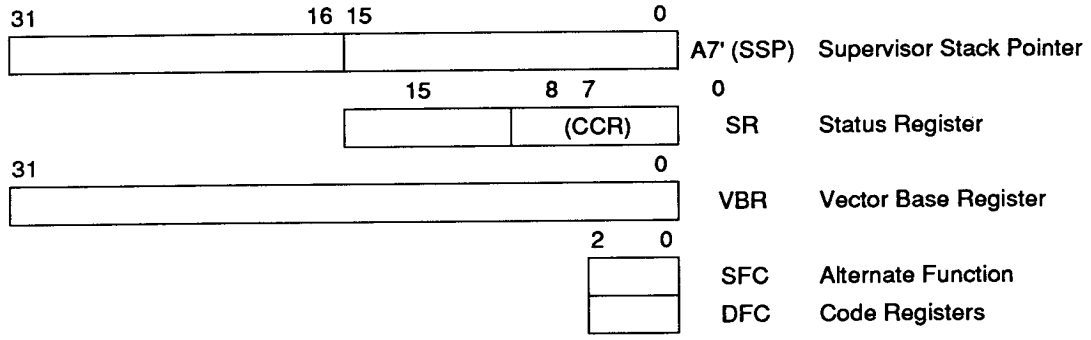


CPU32 Block Diagram

User Programming Model



Supervisor Programming Model Supplement



SR — Status Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
T1	T0	S	0	0	I2	I1	I0	0	0	0	X	N	Z	V	C

Status Register

System Byte
- T1–T0 — Trace Enable
- S — Supervisor/User State
- Bits 12–11 — Unimplemented
- I2–I0 — Interrupt Priority Mask

User Byte (Condition Code Register)
- Bits 7–5 — Unimplemented
- X — Extend
- N — Negative
- Z — Zero
- V — Overflow
- C — Carry

Instruction Set Summary

Mnemonic	Description
ABCD	Add Decimal with Extend
ADD	Add
ADDA	Add Address
ADDI	Add Immediate
ADDQ	Add Quick
ADDX	Add with Extend
AND	Logical AND
ANDI	Logical AND Immediate
ASL, ASR	Arithmetic Shift Left and Right
Bcc	Branch Conditionally
BCHG	Test Bit and Change
BCLR	Test Bit and Clear
BGND	Background
BKPT	Breakpoint
BRA	Branch
BSET	Test Bit and Set
BSR	Branch to Subroutine
BTST	Test Bit
CHK, CHK2	Check Register Against Upper and Lower Bounds
CLR	Clear
CMP	Compare
CMPA	Compare Address
CMPI	Compare Immediate
CMPM	Compare Memory to Memory
CMP2	Compare Register Against Upper and Lower Bounds
DBcc	Test Condition, Decrement and Branch
DIVS, DIVSL	Signed Divide
DIVU, DIVUL	Unsigned Divide
EOR	Logical Exclusive OR
EORI	Logical Exclusive OR Immediate
EXG	Exchange Registers
EXT, EXTB	Sign Extend
LEA	Load Effective Address
LINK	Link and Allocate
LPSTOP	Low Power Stop
LSL, LSR	Logical Shift Left and Right
ILLEGAL	Take Illegal Instruction Trap
JMP	Jump
JSR	Jump to Subroutine

Mnemonic	Description
MOVE	Move
MOVE CCR	Move Condition Code Register
MOVE SR	Move Status Register
MOVE USP	Move User Stack Pointer
MOVEA	Move Address
MOVEC	Move Control Register
MOVEM	Move Multiple Registers
MOVEP	Move Peripheral
MOVEQ	Move Quick
MOVES	Move Alternate Address Space
MULS, MULS.L	Signed Multiply
MULU, MULU.L	Unsigned Multiply
NBCD	Negate Decimal with Extend
NEG	Negate
NEGX	Negate with Extend
NOP	No Operation
OR	Logical Inclusive OR
ORI	Logical Inclusive OR Immediate
PEA	Push Effective Address
RESET	Reset External Devices
ROL, ROR	Rotate Left and Right
ROXL, ROXR	Rotate with Extend Left and Right
RTD	Return and Deallocate
RTE	Return from Exception
RTR	Return and Restore Codes
RTS	Return from Subroutine
SBCD	Subtract Decimal with Extend
Scc	Set Conditionally
STOP	Stop
SUB	Subtract
SUBA	Subtract Address
SUBI	Subtract Immediate
SUBQ	Subtract Quick
SUBX	Subtract with Extend
SWAP	Swap Register Words
TBLS,TBLSN	Table Lookup and Interpolate (Signed)
TBLU, TBLUN	Table Lookup and Interpolate (Unsigned)
TAS	Test Operand and Set
TRAP	Trap
TRAPcc	Trap Conditionally
TRAPV	Trap on Overflow
TST	Test Operand
UNLK	Unlink

Background Mode Command Summary

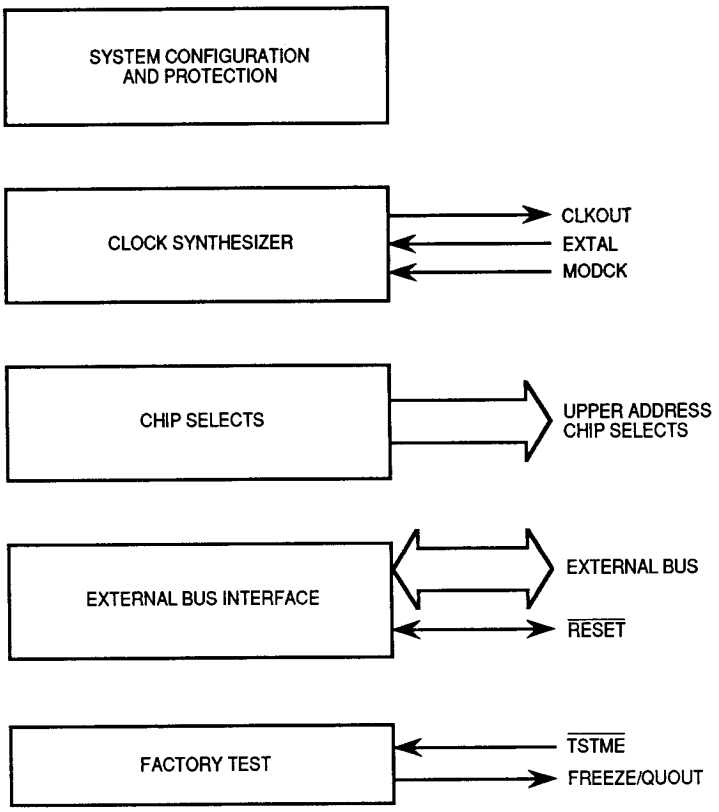
Command	Mnemonic	Description
Read D/A Register	RDREG/RAREG	Read the selected address or data register and return the results through the serial interface.
Write D/A Register	WDREG/WAREG	The data operand is written to the specified address or data register.
Read System Register	RSREG	The specified system control register is read. All registers that can be read in supervisor mode can be read in background mode.
Write System Register	WSREG	The operand data is written into the specified system control register.
Read Memory Location	READ	Read the sized data at the memory location specified by the long-word address. The source function code register (SFC) determines the address space accessed.
Write Memory Location	WRITE	Write the operand data to the memory location specified by the long-word address. The destination function code (DFC) register determines the address space accessed.
Dump Memory Block	DUMP	Used in conjunction with the READ command to dump large blocks of memory. An initial READ is executed to set up the starting address of the block and retrieve the first result. Subsequent operands are retrieved with the DUMP command.
Fill Memory Block	FILL	Used in conjunction with the WRITE command to fill large blocks of memory. Initially,a WRITE is executed to set up the starting address of the block and supply the first operand. The FILL command writes subsequent operands.
Resume Execution	GO	The pipe is flushed and refilled before resuming instruction execution at the current PC.
Patch User Code	CALL	Current program counter is stacked at the location of the current stack pointer. Instruction execution begins at user patch code.
Reset Peripherals	RST	Asserts RESET for 512 clock cycles. The CPU is NOT reset by this command. Synonymous with the CPU RESET instruction.
No Operation	NOP	NOP performs no operation and can be used as a null command.

System Integration Module (SIM)

The MC68332 system integration module (SIM) consists of five submodules that control the microcontroller unit (MCU) system startup, initialization, configuration, and external bus. This high level of integration requires very little or no external glue logic, as was necessary with MPU architectures of the past.

Features

- System Configuration and Protection
- Clock Synthesizer
- Chip Selects
- External Bus Interface



System Integration Module (SIM) Block Diagram

System Configuration and Protection Submodule

This MCU is designed to provide maximum system safeguards and incorporates many of the functions that normally must be provided in external circuits.

Features

- System Configuration

  The module configuration register allows the user to configure the system according to the particular system requirements, such as enable slave mode, bus monitor, changing the module map, etc.
- Internal Bus Monitor

  The MCU provides an internal bus monitor to check the $\overline{\text{DSACKx}}$ response time for all internal bus accesses and internally initiated external bus accesses.
- Halt Monitor

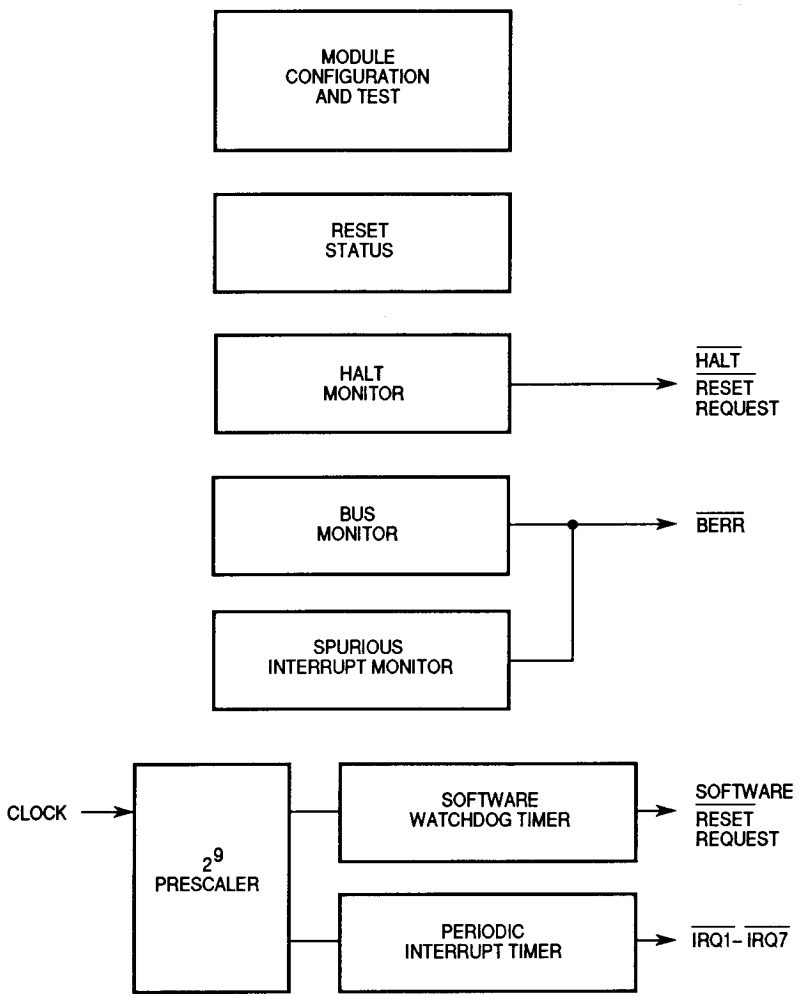
  A halt monitor causes a reset to occur if the CPU asserts an internal halt ($\overline{\text{HALT}}$).
- Spurious Interrupt Monitor

  If no interrupt arbitration occurs during an interrupt acknowledge (IACK) cycle, the $\overline{\text{BERR}}$ signal is asserted internally.
- Software Watchdog

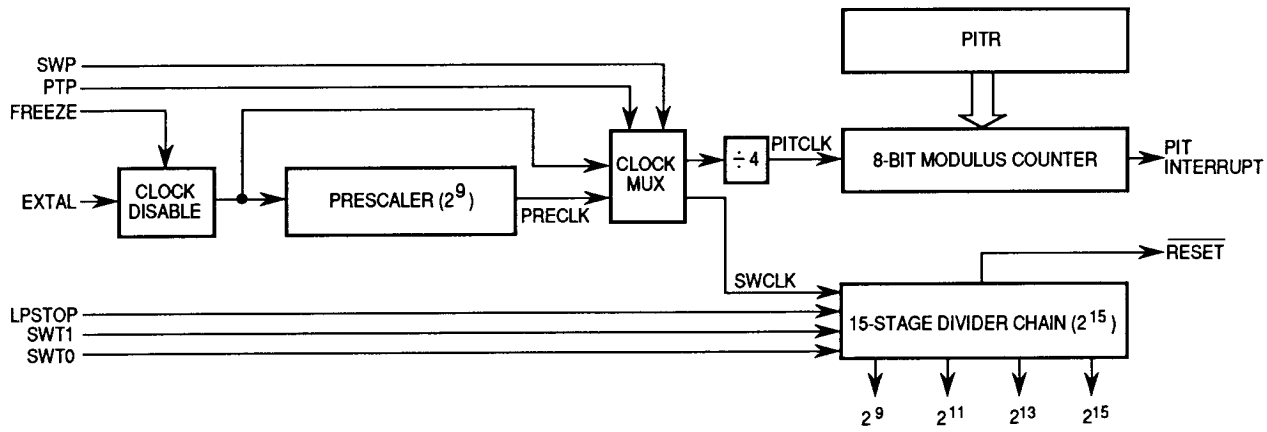
  The watchdog asserts reset if the software fails to service the software watchdog for a designated period of time (presumably because it is trapped in a loop or lost). There are four selectable timeout periods. A prescaler can be used for long timeout periods. The software watchdog is automatically enabled after reset.
- Periodic Interrupt Timer

  The MCU provides a timer to generate periodic interrupts, which can vary from 122 µs to15.94 s when using a 32.768-kHz crystal to generate the system clock.
- Reset Status Register

  The reset status register shows the source of the last reset.



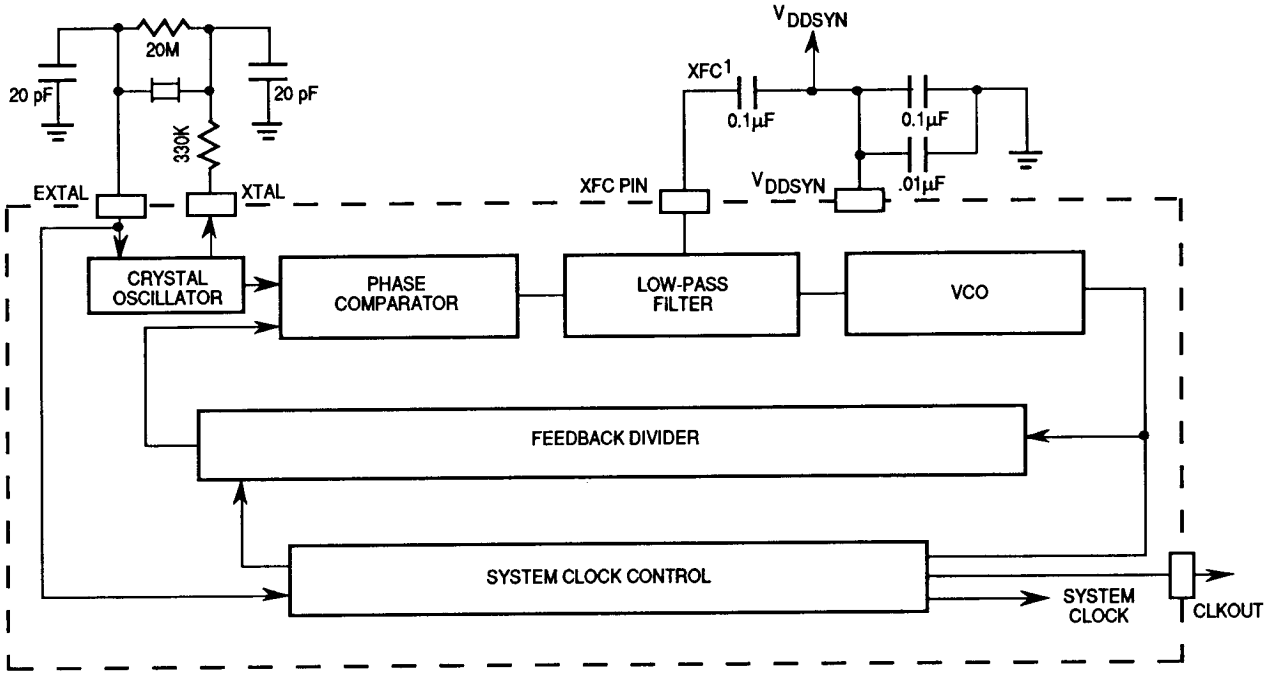
System Configuration and Protection Submodule



Watchdog Timer

Clock Synthesizer (System Clock)

The system clock, generated by an on-chip phase-locked-loop circuit, runs the device up to 16.78 MHz (current maximum operating frequency) from a 32.768-kHz watch crystal. The system speed can be changed dynamically under software control. With its fully static CMOS design, it is possible to stop the system clock completely, by using a low-power stop instruction, and still retain the contents of the registers and on-board RAM.



Notes:
1. Must be low-leakage capacitor.
EXTAL can be driven with an external oscillator.

Clock Submodule Block Diagram

Chip Selects

Typical microcomputer systems require additional hardware to provide peripheral-select signals to external devices. The MC68332 integrates these functions on-chip, providing 12 independent programmable chip selects that can be set up as output enables, read or write strobes, or IACK signals. Block size is programmable from a minimum of 2K bytes to 1M byte in length. Accesses can be preselected for either 8- or 16-bit transfers. Up to 13 wait states can be programmed for insertion during the access. All bus interface signals are handled automatically by the chip-select logic.

Features

- Twelve Programmable Chip-Select Circuits
  Twelve chip-select signals are available ($\overline{\text{CSBOOT}}$ and CS10–CS0). The $\overline{\text{CSBOOT}}$ pin performs a single function because it functions after a reset with no initialization. The other chip-select circuits share functions on their output pins. All 12 chip select circuits are independently programmable from the same list of selectable features.
- Variable Block Sizes
  The block size, starting from the specified base address, can be programmed as 2K, 8K, 16K, 64K, 128K, 256K, 512K bytes or 1M byte.
- Both 8-Bit and 16-Bit Ports Supported
  Eight-bit ports are accessible on both odd and even addresses when connected to data bus bits 15–8. Sixteen-bit ports can be accessed as odd bytes, even bytes, or as words.

- Read Only, Write Only, or Read/Write Capability

  Chip selects, asserted in synchronization with read, write, or both read and write, are possible.
- Address Strobe and Data Strobe Timing Option

  Chip-select signals can be synchronized with either address strobe or data strobe, so that control signals such as output enable or write enable can be generated easily.
- Internal DSACK Generation with Wait States

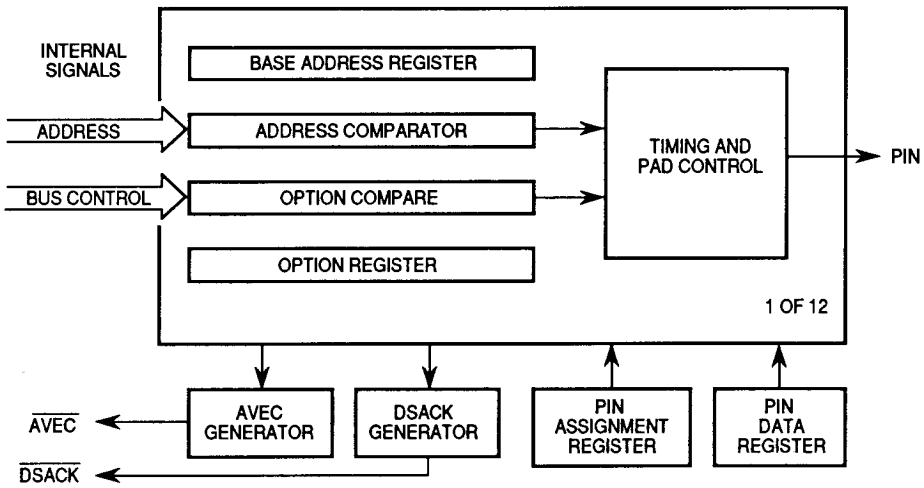
  The port size programmed in the pin assignment register can be referenced for generating DSACK and the proper number of wait states for a particular device programmed by the user.
- Address Space Checking

  Supervisor, User, and CPU space accesses can be checked optionally.
- Interrupt Priority Level Checking

  In the IACK cycle, the acknowledged interrupt level can be compared with the user-specified level programmed in the option field. If autovector option is selected, AVEC is asserted internally.
- Discrete Output

  Port C pins A22–A19 and FC2–FC0 can be programmed for discrete output.
- M6800-Type Peripheral Support

  M6800-type peripherals that require an E clock for synchronization can be supported. Chip select is asserted, synchronized with the E clock on pin A23, providing correct data bus timing for the MCU.



Chip-Select Circuit Block Diagram

External Bus Interface

The external bus, based on the MC68020, provides 24 address lines and a 16-bit data bus. This data bus allows dynamic sizing between 8- and 16-bit data accesses. Read-modify-write cycles are provided through the RMC signal. A three-line handshaking interface accomplishes external bus arbitration.

The external bus interface provides the normal interface between the external system and the internal modules of the MCU and provides alternate functions and discrete I/O capability for many of the pins.The alternate functions of the chip-select circuitry can be determined by the state of the data bus pins at reset, or configured by pin assignment registers.

Refer to the system block diagram and the register descriptions for more information on alternate pin functions.

SIM Register Map (Sheet 1 of 2)

FC	ADDRESS	15	8 7	0	SUBMODULE
101	YFFA00	MODULE CONFIGURATION (MCR)			
101	YFFA02	MODULE TEST (SIMTR)			TEST
101	YFFA04	CLOCK SYNTHESIZER CONTROL (SYNCR)			CLOCK
101	YFFA06	UNUSED	RESET STATUS REGISTER (RSR)		EBI
101	YFFA08	MODULE TEST E (SIMTRE)			TEST
101	YFFA0A	UNUSED	UNUSED		
101	YFFA0C	UNUSED	UNUSED		
101	YFFA0E	UNUSED	UNUSED		
X01	YFFA10	UNUSED	PORTE DATA (PORTE)		EBI
X01	YFFA12	UNUSED	PORTE DATA (PORTE)		
X01	YFFA14	UNUSED	PORTE DATA DIRECTION (DDRE)		
101	YFFA16	UNUSED	PORTE PIN ASSIGNMENT (PEPAR)		
X01	YFFA18	UNUSED	PORTF DATA (PORTF)		
X01	YFFA1A	UNUSED	PORTF DATA (PORTF)		
X01	YFFA1C	UNUSED	PORTF DATA DIRECTION (DDRF)		
101	YFFA1E	UNUSED	PORTF PIN ASSIGNMENT (PFPAR)		EBI
101	YFFA20	UNUSED	SYSTEM PROTECTION CONTROL (SYPCR)		SYS PROTECT
101	YYFA22	PERIODIC INTERRUPT CONTROL (PICR)			MOD CONF
101	YFFA24	PERIODIC INTERRUPT TIMING (PITR)			MOD CONF
101	YFFA26	UNUSED	SOFTWARE SERVICE (SWSR)		SYS PROTECT
101	YFFA28	UNUSED	UNUSED		
101	YFFA30	TEST MODULE MASTER SHIFT A (TSTMSRA)			TEST
101	YFFA32	TEST MODULE MASTER SHIFT B (TSTMSRB)			
101	YFFA34	TEST MODULE SHIFT COUNT.A (TSTSCA)	TEST MODULE SHIFT COUNT.B (TSTSCB)		
101	YFFA36	TEST MODULE REPETITION COUNTER (TSTRC)			
101	YFFA38	TEST MODULE CONTROL (CREG)			
X01	YFFA3A	TEST MODULE DISTRIBUTED REGISTER (DREG)			TEST
	YFFA3C	UNUSED	UNUSED		
	YFFA3E	UNUSED	UNUSED		
X01	YFFA40	UNUSED	PORT C DATA (CSPDR)		CHIP SELECT
X01	YFFA42	UNUSED	UNUSED		
101	YFFA44	CHIP-SELECT PIN ASSIGNMENT (CSPAR0)			
101	YFFA46	CHIP-SELECT PIN ASSIGNMENT (CSPAR1)			
101	YFFA48	CHIP-SELECT BASE BOOT (CSBARBT)			
101	YFFA4A	CHIP-SELECT OPTION BOOT (CSORBT)			
101	YFFA4C	CHIP-SELECT BASE 0 (CSBAR0)			
101	YFFA4E	CHIP-SELECT OPTION 0 (CSOR0)			
101	YFFA50	CHIP-SELECT BASE 1 (CSBAR1)			
101	YFFA52	CHIP-SELECT OPTION 1 (CSOR1)			
101	YFFA54	CHIP-SELECT BASE 2 (CSBAR2)			
101	YFFA56	CHIP-SELECT OPTION 2 (CSOR2)			
101	YFFA58	CHIP-SELECT BASE 3 (CSBAR3)			
101	YFFA5A	CHIP-SELECT OPTION 3 (CSOR3)			
101	YFFA5C	CHIP-SELECT BASE 4 (CSBAR4)			
101	YFFA5E	CHIP-SELECT OPTION 4 (CSOR4)			CHIP SELECT

SIM Register Map (Sheet 2 of 2)

FC	ADDRESS	15 8 7 0	SUBMODULE
101	YFFA60	CHIP-SELECT BASE 5 (CSBAR5)	CHIP SELECT
101	YFFA62	CHIP-SELECT OPTION 5 (CSOR5)	
101	YFFA64	CHIP-SELECT BASE 6 (CSBAR6)	
101	YFFA66	CHIP-SELECT OPTION 6 (CSOR6)	
101	YFFA68	CHIP-SELECT BASE 7 (CSBAR7)	
101	YFFA6A	CHIP-SELECT OPTION 7 (CSOR7)	
101	YFFA6C	CHIP-SELECT BASE 8 (CSBAR8)	
101	YFFA6E	CHIP-SELECT OPTION 8 (CSOR8)	
101	YFFA70	CHIP-SELECT BASE 9 (CSBAR9)	
101	YFFA72	CHIP-SELECT OPTION 9 (CSOR9)	
101	YFFA72	CHIP-SELECT OPTION 9 (CSOR9)	
101	YYFA74	CHIP-SELECT BASE 10 (CSBAR10)	
101	YFFA76	CHIP SELECT OPTION 10 (CSOR10)	CHIP SELECT
	YFFA78	UNUSED / UNUSED	
	YFFA7A	UNUSED / UNUSED	
	YFFA7C	UNUSED / UNUSED	
	YFFA7E	UNUSED / UNUSED	

X = Depends on state of SUPV bit in SIM MCR
Y = m111, where m is the modmap bit in the SIM MCR (Y = $7 or $F)

SIM Registers

MCR — Module Control Register **$YFFA00**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXOFF	FRZSW	FRZBM	0	SLVEN	0	SHEN1	SHEN0	SUPV	MM	0	0	IARB3	IARB2	IARB1	IARB0
RESET:															
0	1	1	0	DB11	0	0	0	1	1	0	0	1	1	1	1

EXOFF — External Clock Off
- 1 = The CLKOUT pin is placed in a high-impedance state.
- 0 = The CLKOUT pin is driven from an internal clock source.

FRZSW — Freeze Software Enable
- 1 = When FREEZE is asserted, the software watchdog and periodic interrupt timer counters are disabled, preventing interrupts from occurring when software is debugged.
- 0 = When FREEZE is asserted, the software watchdog and periodic interrupt timer counters continue to run.

FRZBM — Freeze Bus Monitor Enable
- 1 = When FREEZE is asserted, the bus monitor is disabled.
- 0 = When FREEZE is asserted, the bus monitor continues to operate as programmed.

SLVEN — Slave Mode Enabled
- 1 = Any external master winning control of the external bus also gains direct access to the internal peripherals.
- 0 = The internal peripherals are not available to an external master. This bit is a read-only status bit that reflects the state of DB11 during reset.

SHEN1–0— Show Cycle Enable

These two control bits determine what the EBI does with the external bus during internal transfer operations.

SHEN1	SHEN0	Action
0	0	Show cycles disabled, external arbitration enabled
0	1	Show cycles enabled, external arbitration disabled
1	0	Show cycles enabled, external arbitration enabled
1	1	Show cycles enabled, external arbitration enabled — internal activity halted by a bus grant

SUPV — Supervisor/Unrestricted Data Space

1 = Registers with access controlled by the SUPV bit are restricted to supervisor access only when FC2 = 1.

0 = Registers with access controlled by the SUPV bit are unrestricted (FC2 is a don't care).

MM — Module Mapping

1 = Internal modules are addressed from $FFF000–$FFFFFF, which is in the absolute short addressing range.

0 = Internal modules are addressed from $7FF000–$7FFFFF.

IARB3–IARB0 — Interrupt Arbitration Bits

The system software must initialize the IARB field to a value from $F (highest priority) to $1 (lowest priority). The reset value of zero keeps the SIM from arbitrating during an IACK cycle and causes any SIM interrupts to be discarded as spurious interrupts.

SIMTR — System Integration Module Test Register **$YYFA02**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MASK						0	0	SOSEL1	SOSEL0	SHIRQ1	SHIRQ0	FBIT1	FBIT0	BWC1	BWC0
RESET:															
0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0

MASK — Revision Number for this Part

SOSEL1–SOSEL0 — Scan Out Select

SHIRQ1–SHIRQ0 — Show Interrupt Request

FBIT1–FBIT0— Force Bits

BWC1–BWC0 — Bandwidth Control Bits

SYNCR — Clock Synthesizer Control Register **$YFFA04**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
W	X	Y5	Y4	Y3	Y2	Y1	Y0	EDIV	0	0	SLIMP	SLOCK	RSTEN	STSIM	STEXT
RESET:															
0	0	1	1	1	1	1	1	0	0	0	U	U	0	0	0

U = Unaffected by reset

$$F_{SYSTEM} = F_{CRYSTAL}\,[4(Y + 1)2^{2W} + X]$$

W — Frequency Control Bit

X — Frequency Control Bit

Y5–Y0 — Frequency Control Bits

EDIV — E-Clock Divide Rate
- 1 = E clock = system clock divided by 16
- 0 = E clock = system clock divided by 8

SLIMP — Limp Mode
- 1 = A loss of crystal reference has been detected and the VCO is running at approximately half of maximum speed.
- 0 = External crystal frequency is VCO reference.

SLOCK — Synthesizer Lock
- 1 = VCO has locked on to the desired frequency (or system clock is driven externally).
- 0 = VCO is enabled, but has not yet locked.

RSTEN — Reset Enable
- 1 = Loss of crystal causes a system reset.
- 0 = Loss of crystal causes the VCO to operate at a nominal speed without external reference (limp mode), and the MCU continues to operate at that speed.

STSIM — Stop Mode System Integration Clock
- 1 = When the LPSTOP instruction is executed, the SIM clock is driven from the VCO.
- 0 = When the LPSTOP instruction is executed, the SIM clock is driven from the crystal oscillator and the VCO is turned off to conserve power.

STEXT — Stop Mode External Clock
- 1 = When the LPSTOP instruction is executed, the external clock pin (CLKOUT) is driven from the SIM clock, as determined by the STSIM bit.
- 0 = When the LPSTOP instruction is executed, the external clock is held low to conserve power.

Some System Frequencies from 32.768-kHz Reference

Y	W = 0 X = 0	W = 0 X = 1	W = 1 X = 0	W = 1 X = 1
0 = 000000	131	262	524	1049
1 = 000001	262	524	1049	2097
2 = 000010	393	786	1573	3146
3 = 000011	524	1049	2097	4194
4 = 000100	655	1311	2621	5243
5 = 000101	786	1573	3146	6291
6 = 000110	918	1835	3670	7340
7 = 000111	1049	2097	4194	8389
8 = 001000	1180	2359	4719	9437
9 = 001001	1311	2621	5243	10486
10 = 001010	1442	2884	5767	11534
11 = 001011	1573	3146	6291	12583
12 = 001100	1704	3408	6816	13631
13 = 001101	1835	3670	7340	14680
14 = 001110	1966	3932	7864	15729
15 = 001111	2097	4194	8389	16777
16 = 010000	2228	4456	8913	

RSR — Reset Status Register **$YFFA07**

7	6	5	4	3	2	1	0
EXT	POW	SW	HLT	0	LOC	SYS	TST

EXT — External Reset
1 = The last reset was caused by an external signal.

POW — Power-Up Reset
1 = The last reset was caused by the power-up reset circuit.

SW — Software Watchdog Reset
1 = The last reset was caused by the software watchdog circuit.

HLT — Halt Monitor Reset
1 = The last reset was caused by the system protection submodule halt monitor.

LOC — Loss of Clock Reset
1 = The last reset was caused by a loss of frequency reference to the clock submodule.

SYS — System Reset
1 = The last reset was caused by the CPU executing a reset instruction.

TST — Test Submodule Reset
1 = The last reset was caused by the test submodule.

SIMTRE — System Integration Module Test Register (E Clock) **YFFA08**

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

RESET:

This write-only register is reserved for factory testing. A write to this register in test mode forces the E-clock phase to synchronize with the sytem clock.

PORTE — Port E Data Register **$YFFA11, YFFA13**

7	6	5	4	3	2	1	0
PE7	PE6	PE5	PE4	PE3	PE2	PE1	PE0
RESET:							
U	U	U	U	U	U	U	U

PE7–0 — Port E Data

DDRE — Port E Data Direction Register **$YFFA15**

7	6	5	4	3	2	1	0
DDE7	DDE6	DDE5	DDE4	DDE3	DDE2	DDE1	DDE0
RESET:							
0	0	0	0	0	0	0	0

DDE7–0 — Data Direction E (Input/Output)
1 = Input
0 = Output

PEPAR — Port E Pin Assignment Register **$YFFA17**

7	6	5	4	3	2	1	0
PEPA7 (SIZ1)	PEPA6 (SIZ0)	PEPA5 ($\overline{AS}$)	PEPA4 ($\overline{DS}$)	PEPA3 ($\overline{RMC}$)	PEPA2 ($\overline{AVEC}$)	PEPA1 $\overline{DSACK1}$	PEPA0 $\overline{DSACK0}$
RESET:							
DB8	DB8	DB8	DB8	DB8	DB8	DB8	DB8

PEPA7–0 — Port E Pin Assignment (I/O Function)
SIZ1–$\overline{DSACK0}$ — Control Bus Function
A one on DB8 at reset sets the pins to the bus control function; otherwise, they are general-purpose I/O.

PORTF — Port F Data Register **$YFFA19, YFFA1B**

7	6	5	4	3	2	1	0
PF7	PF6	PF5	PF4	PF3	PF2	PF1	PF0
RESET:							
U	U	U	U	U	U	U	U

PF7–0 — Port F Data

DDRF — Port F Data Direction Register **$YFFA1D**

7	6	5	4	3	2	1	0
DDF7	DDF6	DDF5	DDF4	DDF3	DDF2	DDF1	DDF0
RESET:							
0	0	0	0	0	0	0	0

DDF7–0 — Data Direction F (Input/Output)
1 = Input
0 = Output

PFPAR — Port F Pin Assignment Register **$YFFA1F**

7	6	5	4	3	2	1	0
PFPA7 ($\overline{IRQ7}$)	PFPA6 ($\overline{IRQ6}$)	PFPA5 ($\overline{IRQ5}$)	PFPA4 ($\overline{IRQ4}$)	PFPA3 ($\overline{IRQ3}$)	PFPA2 ($\overline{IRQ2}$)	PFPA1 ($\overline{IRQ1}$)	PFPA0 (MODCK)
RESET:							
DB9	DB9	DB9	DB9	DB9	DB9	DB9	DB9

PFPA7–0 — Port E Pin Assignment (I/O Function)
IRQ7–1 — Control Bus Function
MODCLK
A one on DB9 at reset sets pins to the bus control function; otherwise, they are general-purpose I/O.

SYPCR — System Protection Control Register **$YFFA21**

7	6	5	4	3	2	1	0
SWE	SWP	SWTI	SWTO	HME	BME	BMT1	BMT0
RESET:							
1	$\overline{MODCK}$	0	0	0	0	0	0

SWE — Software Watchdog Enable
1 = Software watchdog enabled
0 = Software watchdog disabled

SWP — Software Watchdog Prescale
1 = Software watchdog clock prescaled by 512
0 = Software watchdog clock not prescaled
At reset SWP takes on the inverted value of the MODCLK pin.

SWT1–SWT0 — Software Watchdog Timing

These bits control the divide ratio used to establish the timeout period for the software watchdog timer. The software timeout period is given by the following equation.

Timeout Period = 1/(EXTAL Frequency/Divide Count)

or

= Divide Count/EXTAL Frequency

Software Timeout Periods for Watchdog Timer

Bits 6-4	Software Timeout Period	32.768-kHz Crystal Period	16.718-MHz External Clock Period
000	2^9/EXTAL Input Frequency	15.6 Milliseconds	30.6 Microseconds
001	2^{11}/EXTAL Input Frequency	62.5 Milliseconds	122.5 Microseconds
010	2^{13}/EXTAL Input Frequency	250 Milliseconds	490 Microseconds
011	2^{15}/EXTAL Input Frequency	1 Second	1.96 Microseconds
100	2^{18}/EXTAL Input Frequency	8 Seconds	15.6 Milliseconds
101	2^{20}/EXTAL Input Frequency	32 Seconds	62.7 Milliseconds
110	2^{22}/EXTAL Input Frequency	128 Seconds	250 Milliseconds
111	2^{24}/EXTAL Input Frequency	512 Seconds	1 Second

HME — Halt Monitor Enable

1 = Enable halt monitor function

0 = Disable halt monitor function

BME — Bus Monitor External Enable

1 = Enable bus monitor function for an internal to external bus cycle.

0 = Disable bus monitor function for an internal to external bus cycle.

BMT — Bus Monitor Timing

Bits 1–0	Bus Monitor Timeout Period
00	64 System Clocks (CLK)
01	32 System Clocks (CLK)
10	16 System Clocks (CLK)
11	8 System Clocks (CLK)

PICR — Periodic Interrupt Control Register **$YFFA22**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	PIRQL2	PIRQL1	PIRQL0	PIV7	PIV6	PIV5	PIV4	PIV3	PIV2	PIV1	PIV0
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1

PIRQL2–0 — Periodic Interrupt Request Level

Bits 10–8	Interrupt Request Level
000	Periodic Interrupt Disabled
001	Interrupt Request Level 1
010	Interrupt Request Level 2
011	Interrupt Request Level 3
100	Interrupt Request Level 4
101	Interrupt Request Level 5
110	Interrupt Request Level 6
111	Interrupt Request Level 7

PIV7–PIV0 — Periodic Interrupt Vector

These bits contain the value of the vector generated during an IACK cycle in response to an interrupt from the periodic timer.

PITR — Periodic Interrupt Timing Register **$YFFA24**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	PTP	PITR7	PITR6	PITR5	PITR4	PITR3	PITR2	PITR1	PITR0
RESET:															
0	0	0	0	0	0	0	$\overline{\text{MODCK}}$	0	0	0	0	0	0	0	0

PTP — Periodic Timer Prescaler

1 = Periodic timer clock prescaled by a value of 512
0 = Periodic timer clock not prescaled

MODCK	PTP
0	1
1	0

PITR7–PITR0 — PITM Field (Periodic Interrupt Timing Modulus)

The periodic interrupt timing register (PITR) contains the count value for the periodic timer. A zero value turns off the periodic timer.

The period of the periodic timer can be calculated using the following equation:

$$\text{PIT Period} = \text{PITM}/(\text{EXTAL}/\text{Prescaler})/4$$

where

PIT period = Periodic interrupt timer period
PITM = Periodic interrupt timer register modulus (PITR7–PITR0)
EXTAL= Crystal frequency
Prescaler = 512 or 1 depending on the state of the PTP bit in the PITR

SWSR — Software Service Register **$YFFA27**

7	6	5	4	3	2	1	0
SWSR7	SWSR6	SWSR5	SWSR4	SWSR3	SWSR2	SWSR1	SWSR0
RESET:							
0	0	0	0	0	0	0	0

The software watchdog service sequence consists of the following two steps:

1. Write $55 to the software service register (SWSR)
2. Write $AA to the SWSR

Both writes must occur in the order shown before the watchdog timeout, but any number of instructions can be executed between the two writes.

TSTMSRA — Master Shift Register A **$YFFA30**

Master shift register A contains the stimulus to be transferred from the test submodule to the module under test.

TSTMSRB — Master Shift Register B **$YFFA32**

Master shift register B collects the response data shifted from the module under test to the test submodule.

TSTSCA — Shift Count Register A and Shift Counter A **$YFFA34**

Shift count register A is an 8-bit shift register that can be accessed by the bus master. Shift counter A is an 8-bit counter that is loaded by shift count register A and is not accessible to the bus master.

TSTSCB — Shift Count Register B and Shift Counter B **$YFFA35**

Shift count register B is an 8-bit shift register that can be accessed by the bus master. Shift counter B is an 8-bit counter that is loaded by shift count register B and is not accessible to the bus master.

TSTRC — Test Module Repetition Counter **$YFFA36**

The reps counter determines the number of psuedorandom vectors generated in the automatic mode of operation.

CREG — Test Submodule Control Register **$YFFA38**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BUSY	TMARM	COMP	IMBTST	CPUTR	QBIT	MUXEL	—	—	—	—	ACUT	SCONT	SSHOP	SATO	ETM
RESET:															
1	$\overline{\text{TSTME}}$	U	0	0	0	0	0	0	0	0	0	0	0	0	0

ETM — Enter Test Mode
- 1 = Enter test mode.
- 0 = Stay in normal mode.

SATO — Start Automatic Test Operation

1 = Start an automatic test operation.
0 = Stay in normal mode.

SSHOP — Start Shifting Operation

1 = Start a shifting operation.
0 = Stay in normal mode.

SCONT — Start Continuous Operation

1 = Start continuous operation.
0 = Stop continuous operation.

ACUT — Activate Circuit Under Test

1 = Assert the ACUTL line.
0 = Stay in normal mode.

MUXSEL — Multiplexer Select Bit

1 = Shift in source for master shift register B (MSRB) is the external interrupt pin.
0 = Shift in source for MSRB is the internal test line.

QBIT — Quotient Bit

1 = The least significant bit of master shift register B is available at the quotient/freeze (FREEZE/QUOT) pin.
0 = The internal freeze status is available at the FREEZE/QUOT pin.

CPUTR — CPU Test Register

1 = Scan lines are connected to the CPU test register.
0 = Scan lines are disconnected from the CPU test register.

IMBTST — Intermodule Bus Test

1 = Internal interconnect lines are configured as test lines.
0 = Internal interconnect lines have normal function.

COMP — Compare Status Bit

1 = Master shift register B contains the correct answer for the user basic self-test.
0 = Master shift register B does not contain the correct answer for the basic user self-test.

TMARM — Test Mode Armed Status Bit

1 = TSTME pin is asserted; test mode can be entered by setting the ETM control bit.
0 = TSTME pin is negated; test mode cannot be entered. This status bit can be read at any time, but cannot be written.

BUSY — Test Submodule Busy Status Bit

1 = Test submodule is busy.
0 = Test submodule is not busy.

DREG — Distributed Register **$YFFA3A**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
—	—	—	—	—	WAIT3	WAIT2	WAIT1	MSRA18	MSRA17	MSRA16	MSRAC	MSRB18	MSRB17	MSRB16	MSRBC
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

WAIT3–WAIT1 — Wait Counter Preset 3–1
These bits program the delay time between automatic test sequences.

MSRA18–MSRA16 — Master Shift Register A Bits 18–16

MSRAC — Master Shift Register A Configuration
1 = Master shift register A is configured as a 19-bit serial pattern generator.
0 = Master shift register A is configured as a 16-bit shift register.

MSRB18–MSRB16 — Master Shift Register B Bits 18–16

MSRBC — Master Shift Register B Configuration
1 = Master shift register B is configured as a 19-bit serial signature analyzer.
0 = Master shift register B is configured as a 16-bit shift register.

CSPDR — Chip-Select Pin Data Register **$YFFA41**

7	6	5	4	3	2	1	0
0	DO6	DO5	DO4	DO3	DO2	DO1	DO0
RESET:							
0	0	0	0	0	0	0	0

D6–0 — Pin Data

CSPAR0 — Chip-Select Pin Assignment Register 0 **$YFFA44**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	CS5	(FC2)	CS4	(FC1)	CS3	(FC0)	CS2 (BGACK)		CS1	(BG)	CS0 (BR)		CSBOOT	
RESET:															
0	0	DB2	1	DB2	1	DB2	1	DB1	1	DB1	1	DB1	1	1	DB0

Bits 15–14 — Not Used
These bits always read zero; write has no effect.

CSPAR1 — Chip-Select Pin Assignment Register 1 **$YFFA46**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	CS10 (A23)		CS9 (A22)		CS8 (A21)		CS7 (A20)		CS6 (A19)	
RESET:															
0	0	0	0	0	0	DB7	1	DB6	1	DB5	1	DB4	1	DB3	1

Bits 15–10 — Not Used

These bits always read zero; write has no effect.

Hierarchical Selection Structure of CSPAR1

Data Bus Pins at Reset					Default/Alternate Function				
DB7	DB6	DB5	DB4	DB3	CS10/A23	CS9/A22	CS8/A21	CS7/A20	CS6/A19
1	1	1	1	1	CS10	CS9	CS8	CS7	CS6
1	1	1	1	0	CS10	CS9	CS8	CS7	A19
1	1	1	0	X	CS10	CS9	CS8	A20	A19
1	1	0	X	X	CS10	CS9	A21	A20	A19
1	0	X	X	X	CS10	A22	A21	A20	A19
0	X	X	X	X	A23	A22	A21	A20	A19

Pin Assignment Register Bit Encoding

Bits	Description
00	Discrete Output (E Clock on A23)*
01	Alternate Function
10	Chip Select (8-Bit Port)
11	Chip Select (16-Bit Port)

*Except for BR, BG, and BGACK

CSBARBT — Chip-Select Base Address Register Boot **$YFFA48**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	BLKSZ		
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1

CSBAR0–CSBAR10 — Chip-Select Base Address Registers **$YFFA4C–YFFA74**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	BLKSZ		
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

BLKSZ — Block Size Field

Block Size Field	Block Size	Address Lines Compared
000	2K	A23–A11
001	8K	A23–A13
010	16K	A23–A14
011	64K	A23–A16
100	128K	A23–A17
101	256K	A23–A18
110	512K	A23–A19
111	1M	A23–A20

Bits 15–3 — Base Address Field
In supervisor/user space, this field sets the starting address of a particular address space.

CSORBT — Chip-Select Option Register **$YFFA4A**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MODE	BYTE		R/$\overline{W}$		STRB	$\overline{DSACK}$				SPACE		IPL			$\overline{AVEC}$
RESET:															
0	1	1	1	1	0	1	1	0	1	1	1	0	0	0	0

Option Register Functions Summary

Mode	Byte	R/$\overline{W}$	STRB	$\overline{DSACK}$	Space	IPL	$\overline{AVEC}$
0 = ASYNC	00 = Off	00 = Rsvd	0 = AS	0000 = 0 WAIT	00 = CPU SP	000 = All	0 = Off
1 = SYNC	01 = Lower	01 = Read	1 = DS	0001 = 1 WAIT	01 = User SP	001 = Level 1	1 = On
	10 = Upper	10 = Write		0010 = 2 WAIT	10 = Supv SP	010 = Level 2	
	11 = Both	11 = Both		0011 = 3 WAIT	11 = S/U SP	011 = Level 3	
				0100 = 4 WAIT		100 = Level 4	
				0101 = 5 WAIT		101 = Level 5	
				0110 = 6 WAIT		110 = Level 6	
				0111 = 7 WAIT		111 = Level 7	
				1000 = 8 WAIT			
				1001 = 9 WAIT			
				1010 = 10 WAIT			
				1011 = 11 WAIT			
				1100 = 12 WAIT			
				1101 = 13 WAIT			
				1110 = F term			
				1111 = External			

CSOR0–CSOR10 — Chip-Select Option Registers **$YFFA4E–YFFA76**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MODE	BYTE		R/$\overline{W}$		STRB	$\overline{DSACK}$				SPACE		IPL			$\overline{AVEC}$
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MODE — Asynchronous/Synchronous Mode
1 = Synchronous mode selected
0 = Asynchronous mode selected

BYTE — Upper/Lower Byte Option
This field is used when the chip-select 16-bit port option in the pin assignment register is selected.

Bits	Description
00	Disable
01	Lower Byte
10	Upper Byte
11	Both Bytes

R/$\overline{W}$ — Read/Write
This option causes the chip select to be asserted only for a read, only for a write, or for both read and write.

Bits	Description
00	Reserved
01	Read Only
10	Write Only
11	Read/Write

STRB — Address Strobe/Data Strobe
1 = Data strobe
0 = Address strobe
This option controls the timing for assertion of a chip select in asynchronous mode.

$\overline{\text{DSACK}}$ — Data Strobe Acknowledge

In asynchronous mode, this option field specifies the source of the $\overline{\text{DSACK}}$ (externally or internally generated).

Bits	Description
0000	No Wait States
0001	1 Wait State
0010	2 Wait States
0011	3 Wait States
0100	4 Wait States
0101	5 Wait States
0110	6 Wait States
0111	7 Wait States
1000	8 Wait States
1001	9 Wait States
1010	10 Wait States
1011	11 Wait States
1100	12 Wait States
1101	13 Wait States
1110	Fast Termination
1111	External $\overline{\text{DSACK}}$

SPACE — Address Space

This option field checks the address spaces indicated by the function codes generated by the CPU.

Bits	Description
00	CPU Space
01	User Space
10	Supervisor Space
11	Supervisor/User Space

IPL — Interrupt Priority Level

In an IACK cycle, the chip-select logic checks the acknowledged interrupt level on address lines A3–A1. If that level matches the level set in the IPL field, then the chip select can be asserted if the match conditions in the other fields are met.

Bits	Description
000	Any Level
001	IPL1
010	IPL2
011	IPL3
100	IPL4
101	IPL5
110	IPL6
111	IPL7

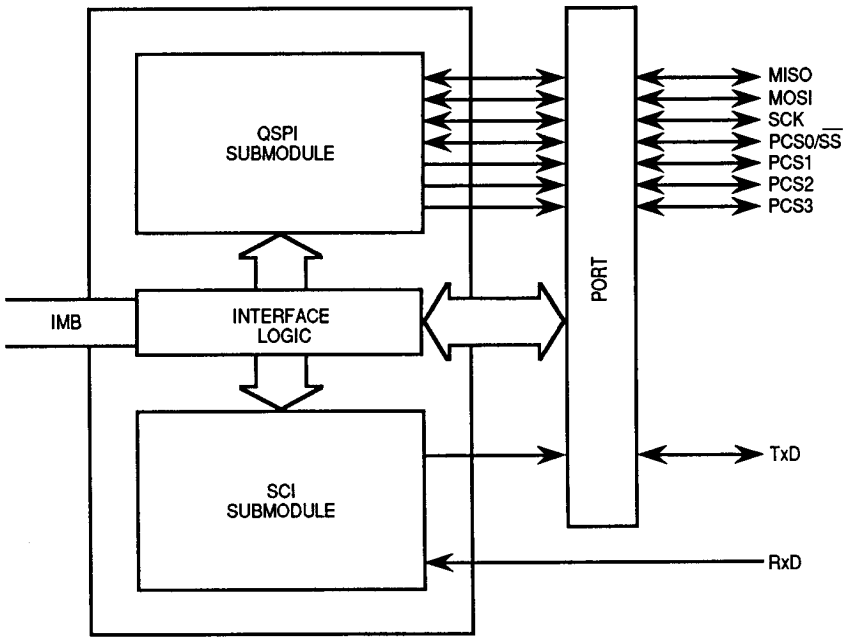
$\overline{\text{AVEC}}$ — Autovector Enable

1 = Autovector enabled

0 = External interrupt vector enabled

Queued Serial Module (QSM)

The queued serial module (QSM) provides the MCU with two serial communication interfaces divided into two submodules: the queued serial peripheral interface (QSPI) and the serial communications interface (SCI). These submodules operate independently.



QSM Block Diagram

QSPI Submodule

The QSPI submodule communicates with external peripherals and other MCUs through a synchronous serial bus. The QSPI is fully compatible with the serial peripheral interface (SPI) found on other Motorola devices, such as the M68HC11 and M68HC05 Families. It has all of the capabilities of the SPI system, as well as several new features.

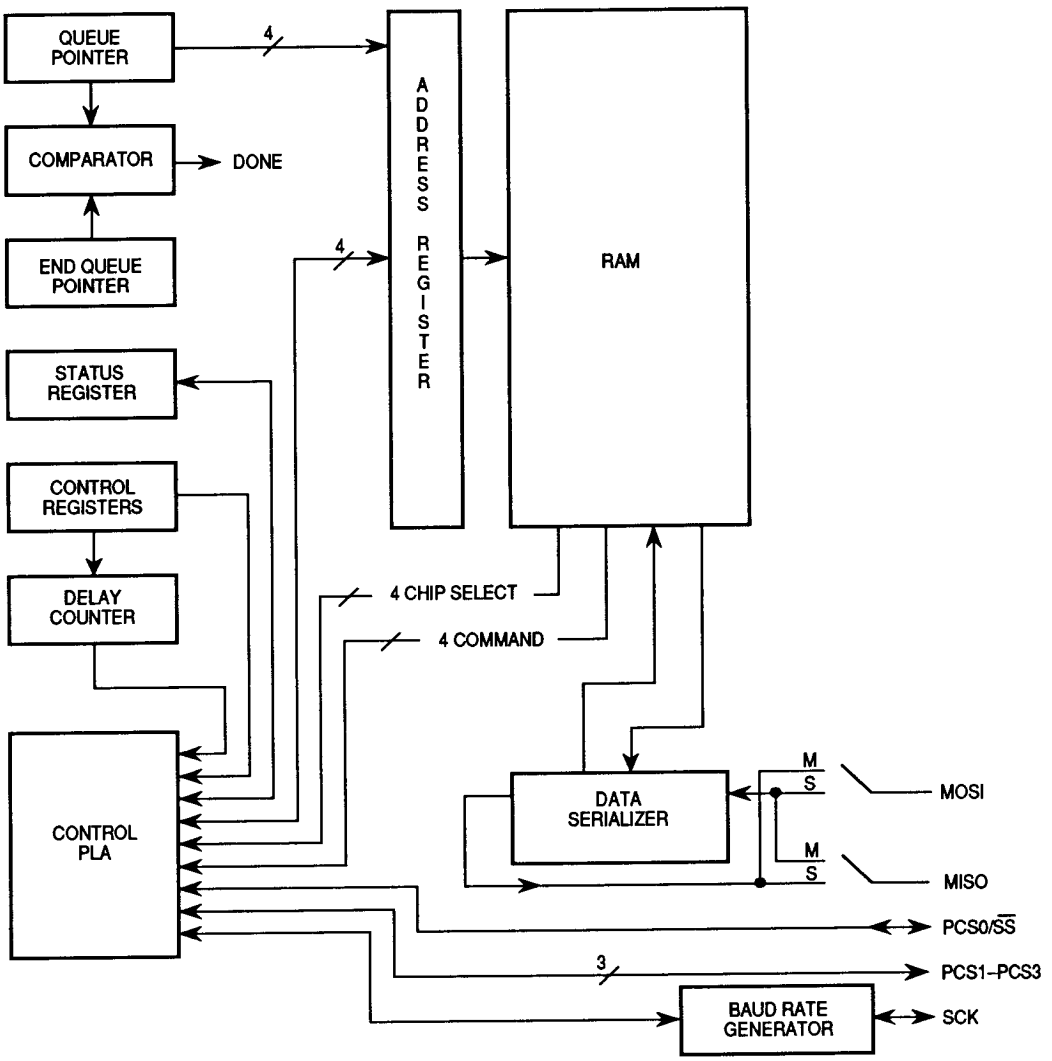
Features

Standard SPI features:

- Full Duplex, Three-Wire Synchronous Transfers
- Half-Duplex, Two-Wire Synchronous Transfers
- Master or Slave Operation
- Programmable Master Bit Rates
- Programmable Clock Polarity and Phase
- End-of-Transmission Interrupt Flag
- Multimaster Mode Fault Flag
- Easy Interface to Simple Expansion Parts (A/D converters, EEPROMS, display drivers, etc.)

QSPI-Enhanced Features:

- Programmable Queue — Up to 16 Preprogrammed Transfers
- Programmable Peripheral Chip Selects — Four Pins Select up to 16 SPI Chips
- Wraparound Transfer Mode — For Autoscanning of Serial A/D (or other) Peripherals
- Programmable Transfer Length — From 8–16 Bits Inclusive
- Programmable Transfer Delay — From 1 µs to 0.5 ms (at 16.78 MHz)
- Programmable Queue Pointer
- Continuous Transfer Mode — Up to 256 Bits



QSPI Submodule Diagram

SCI Submodule

The SCI submodule communicates with external devices and other MCUs through an asynchronous serial bus. The SCI is fully compatible with the SCI systems found on other Motorola MCUs such as the M68HC11 and the M68HC05 Families. It has all of the capabilities of previous SCI systems as well as several significant new features.

Features

Standard SCI Two-Wire System Features:

- Standard Nonreturn-to-Zero (NRZ) Mark/Space Format
- Advanced Error Detection Mechanism (detects noise duration up to 1/16 of a bit time)
- Full-Duplex Operation
- Software Selectable Word Length (8- or 9-bit words)
- Separate Transmitter and Receiver Enable Bits
- May be Interrupt Driven
- Four Separate Interrupt Enable Bits

Standard SCI Receiver Features:

- Receiver Wakeup Function (idle or address mark bit)
- Idle-Line Detect
- Framing Error Detect
- Noise Detect
- Overrun Detect
- Receive Data Register Full Flag

Standard SCI Transmitter Features:

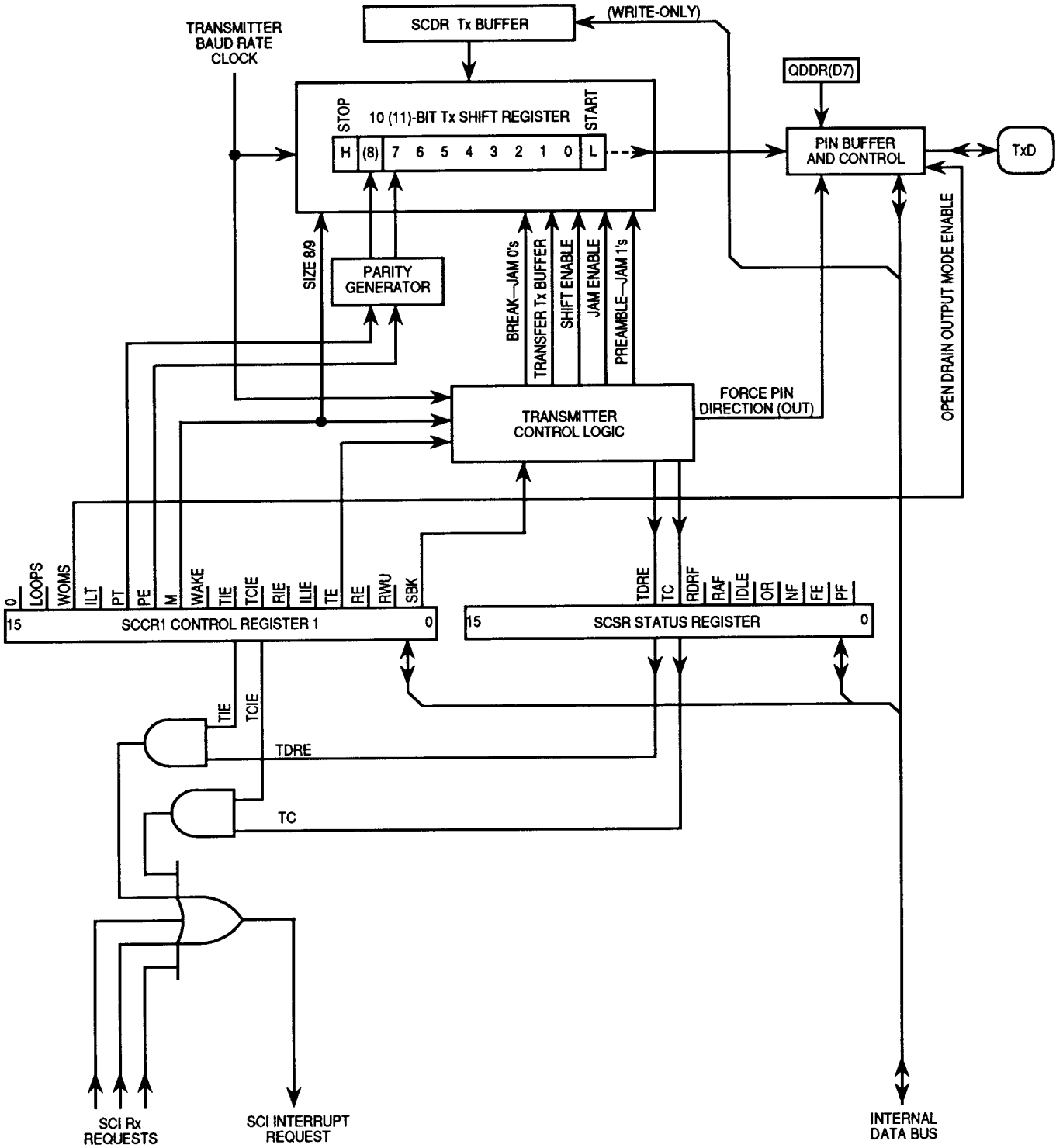
- Transmit Data Register Empty Flag
- Transmit Complete Flag
- Send Break

QSM-Enhanced SCI Two-Wire System Features:

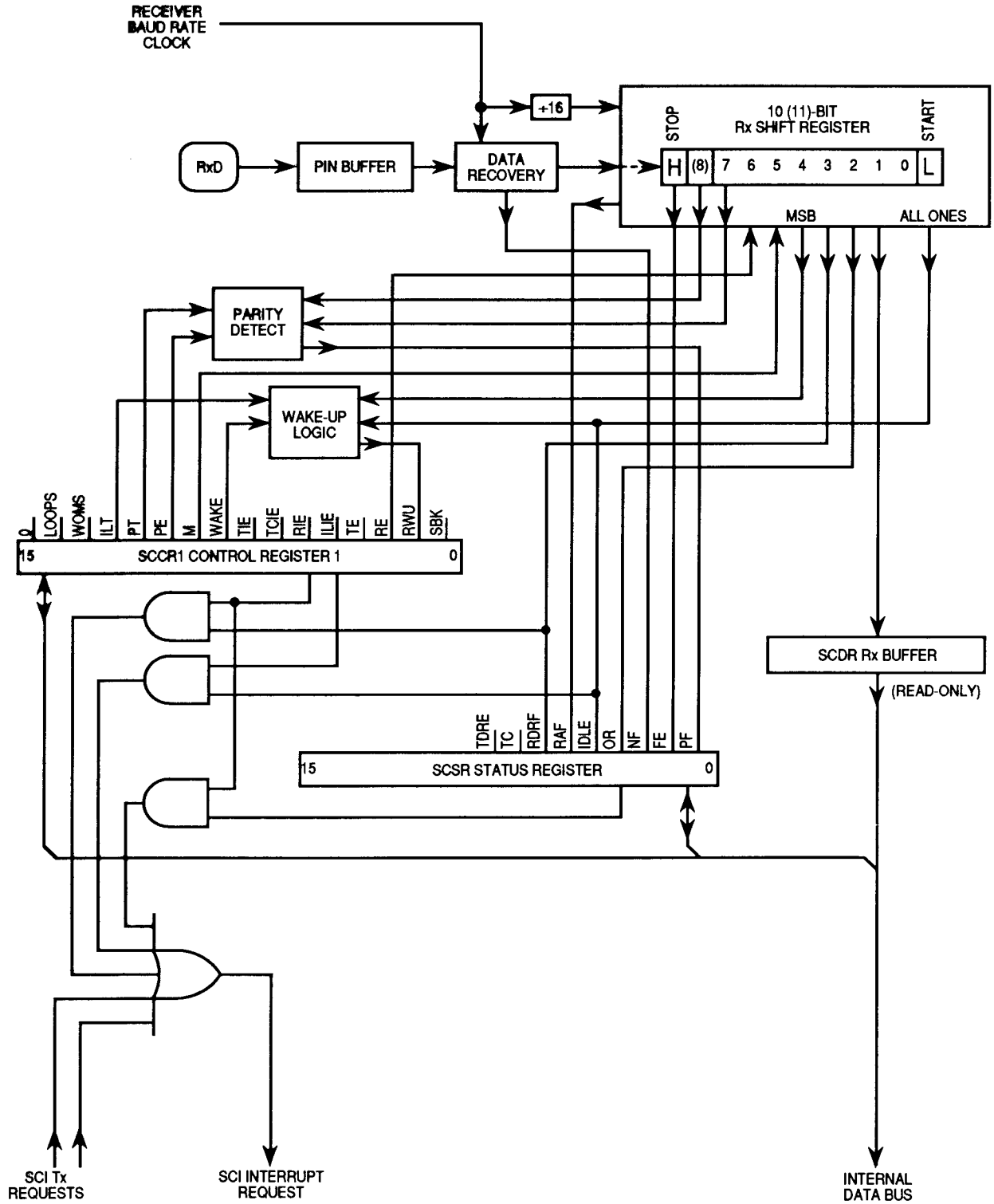
- 13-Bit Programmable Baud Rate Modulus Counter
- Even/Odd Parity Generation and Detection

QSM-Enhanced SCI Receiver Features:

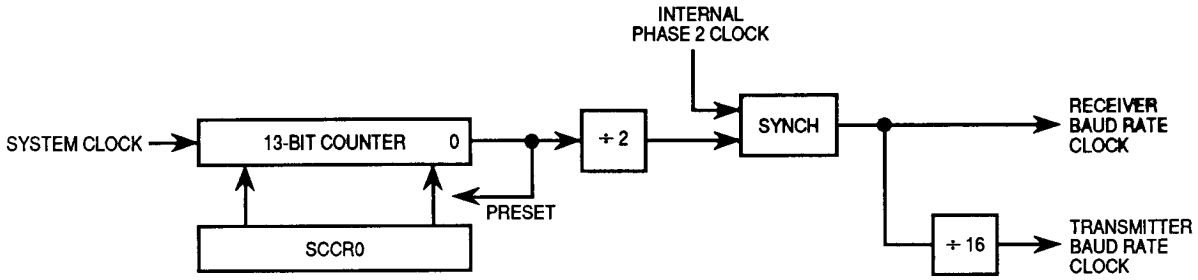
- Two Idle-Line Detect Modes
- Receiver Active Flag
- 13-Bit Programmable Baud Rate Modulus Counter



SCI Transmitter Block Diagram



SCI Receiver Block Diagram



SCI Baud Generator Block Diagram

QSM Memory Map

	15 8	7 0
$YFFC00	QMCR	
$YFFC02	QTEST	
$YFFC04	QILR	QIVR
$YFFC06	RESERVED	
$YFFC08	SCCR0	
$YFFC0A	SCCR1	
$YFFC0C	SCSR	
$YFFC0E	SCDR	
$YFFC10	RESERVED	
$YFFC12	RESERVED	
$YFFC14	RESERVED	QPDR
$YFFC16	QPAR	QDDR
$YFFC18	SPCR0	
$YFFC1A	SPCR1	
$YFFC1C	SPCR2	
$YFFC1E	SPCR3	SPSR
$YFFC20-FF	RESERVED	
$YFFD00-1F	REC.RAM	
$YFFD20-3F	TRAN.RAM	
$YFFD40-4F	COMD.RAM	

SUPERVISOR-ONLY DATA SPACE

ASSIGNABLE DATA SPACE
(SUPERVISOR-ONLY OR UNRESTRICTED)

QUEUE RAM

Y = m111 where m is the modmap bit in the SIM MCR (Y = $7 or $F).

QMCR — QSM Configuration Register **$YFFC00**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STOP	FRZ1	FRZ0	0	0	0	0	0	SUPV	0	0	0	IARB			
RESET:															
0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0

STOP — Stop Enable
1 = QSM clock operation stopped
0 = Normal QSM clock operation

FRZ1 — Freeze 1
1 = Halt the QSM (on a transfer boundary)
0 = Ignore the FREEZE signal on the IMB

FRZ0 — Freeze 0
Reserved for future enhancement.

Bits 12–8 — Not Implemented

SUPV — Supervisor/Unrestricted
1 = Supervisor access
0 = User access

Bits 6–4 — Not Implemented

IARB — Interrupt Arbitration Identification Number
System software should initialize the IARB field to a value between $F (top priority) and $1 (lowest priority). Otherwise, any interrupts generated are identified by the CPU as spurious.

QTEST — QSM Test Register **$YFFC02**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	TSBD	SYNC	TQSM	TMM
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TSBD — SPI Test Scan Path Select
1 = Enable delay to SCK scan path
0 = Enable SPI baud clock scan path

SYNC — SCI Baud Clock Synchronization Signal
1 = Inhibit SCI source signal (QCSCI1)
0 = Activate SCI source signal

TQSM — QSM Test Enable
1 = Enable QSM to send test scan paths
0 = Disable scan path

TMM — Test Memory Map
1 = QSM responds to test memory addresses
0 = QSM responds to QSM memory addresses

QILR — QSM Interrupt Level Register **$YFFC04**

15	14	13	12	11	10	9	8	0
0	0	ILQSPI			ILSCI			XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
RESET:								
0	0	0	0	0	0	0	0	

ILQSPI — Interrupt Level for QSPI
ILQSPI determines the priority level of all QSPI interrupts. Program this field to a value between $0 (interrupts disabled) to $7 (highest priority).

ILSCI — Interrupt Level of SCI
LSCI determines the priority level of all SCI interrupts. Program this field to a value between $0 (interrupts disabled) to $7 (highest priority).

QIVR — QSM Interrupt Vector Register **$YFFC05**

15	7	6	5	4	3	2	1	0
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	INTV							
RESET:								
	0	0	0	0	1	1	1	1

INTV — Interrupt Vector

SCCR0 — SCI Control Register 0 **$YFFC08**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	SCBR												
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0

Bits 15–13 — Not Implemented

SCBR — Baud Rate
The SCI baud rate is programmed by writing a 13-bit value to SCBR.

$$\text{SCI Baud} = \text{System Clock}/(32 \bullet \text{SCBR})$$

where SCBR equals {1, 2, 3, . . . 8191}.

SCCR1 — SCI Control Register 1 **$YFFC0A**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	LOOPS	WOMS	ILT	PT	PE	M	WAKE	TIE	TCIE	RIE	ILIE	TE	RE	RWU	SBK
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit 15 — Not Implemented

LOOPS — LOOP Mode

1 = Test SCI operation, looping, feedback path enabled
0 = Normal SCI operation, no looping, feedback path disabled

LOOPS controls a feedback path on the data serial shifter.

WOMS — Wired-OR Mode for SCI Pins

1 = If configured as an output, TXD is an open-drain output.
0 = If configured as an output, TXD is a normal CMOS output.

ILT — Idle-Line Detect Type

1 = Long idle-line detect (starts counting when the first one is received after a stop bit(s))
0 = Short idle-line detect (starts counting when the first one is received)

PT — Parity Type

1 = Odd parity
0 = Even parity

PE — Parity Enable

1 = SCI parity enabled
0 = SCI parity disabled

M	PE	Result
0	0	8 Data Bits
0	1	7 Data Bits, 1 Parity Bit
1	0	9 Data Bits
1	1	8 Data Bits, 1 Parity Bit

M — Mode Select

1 = SCI frame: 1 start bit, 9 data bits, 1 stop bit (11 bits total)
0 = SCI frame: 1 start bit, 8 data bits, 1 stop bit (10 bits total)

WAKE — Wakeup by Address Mark

1 = SCI receiver awakened by address mark (eighth or ninth (last) bit set)
0 = SCI receiver awakened by idle-line detection

TIE — Transmit Interrupt Enable

1 = SCI TDRE interrupts enabled
0 = SCI TDRE interrupts inhibited

TCIE — Transmit Complete Interrupt Enable

1 = SCI TC interrupts enabled
0 = SCI TC interrupts inhibited

RIE — Receiver Interrupt Enable
1 = SCI RDRF interrupts enabled
0 = SCI RDRF interrupts inhibited

ILIE — Idle-Line Interrupt Enable
1 = SCI IDLE interrupts enabled
0 = SCI IDLE interrupts inhibited

TE — Transmitter Enable
1 = SCI transmitter enabled; TXD pin dedicated to the SCI transmitter
0 = SCI transmitter disabled; TXD pin can be used as general-purpose I/O

RE — Receiver Enable
1 = SCI receiver enabled
0 = SCI receiver disabled

RWU — Receiver Wakeup
1 = Wakeup mode enabled, all received data ignored until awakened
0 = Normal receiver operation, all received data recognized

SBK — Send Break
1 = Break frame(s) transmitted after completion of the current frame
0 = Normal operation

SCSR — SCI Status Register **$YFFC0C**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	TDRE	TC	RDRF	RAF	IDLE	OR	NF	FE	PF
RESET:															
0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0

Bits 15–9 — Not Implemented

TDRE — Transmit Data Register Empty Flag
1 = A new character can now be written to register TDR.
0 = Register TDR still contains data to be sent to the transmit serial shifter.

TC — Transmit Complete Flag
1 = SCI transmitter is idle.
0 = SCI transmitter is busy.

RDRF — Receive Data Register Full Flag
1 = Register RDR contains new data.
0 = Register RDR is empty or contains previously read data.

RAF — Receiver Active Flag
1 = SCI receiver is busy.
0 = SCI receiver is idle.

IDLE — Idle-Line Detected Flag
1 = SCI receiver detected an idle-line condition.
0 = SCI receiver did not detect an idle-line condition.

OR — Overrun Error Flag
1 = RDRF is not cleared before new data arrives.
0 = RDRF is cleared before new data arrives.

NF — Noise Error Flag
1 = Noise occurred on the received data.
0 = No noise was detected on the received data.

FE — Framing Error Flag
1 = Framing error or break occurred on the received data.
0 = No framing error on the received data.

PF — Parity Error Flag
1 = Parity error occurred on the received data.
0 = No parity error on the received data.

SCDR — SCI Data Register **$YFFC0E**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	R8/T8	R7/T7	R6/T6	R5/T5	R4/T4	R3/T3	R2/T2	R1/T1	R0/T0
RESET:															
0	0	0	0	0	0	0	U	U	U	U	U	U	U	U	U

R8/T8 — Receive 8/Transmit 8
R0–R7/T0–T7 — Receive 0–7/Transmit 0–7

QPDR — QSM Port Data Register **$YFFC15**

15	7	6	5	4	3	2	1	0
RESERVED	D7 (TXD)	D6 (PCS3)	D5 (PCS2)	D4 (PCS1)	D3 (PCS0/$\overline{SS}$)	D2 (SCK)	D1 (MOSI)	D0 (MISO)
RESET:								
	0	0	0	0	0	0	0	0

D7–0 — Pin Data
TXD–MIS0 — Pin Function

QPAR — QSM Pin Assignment Register **$YFFC16**

15	14	13	12	11	10	9	8	0
0	PCS3	PCS2	PCS1	PCSO/$\overline{SS}$	0	MOSI	MISO	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
RESET:								
0	0	0	0	0	0	0	0	

0 = General-purpose I/O
1 = QSPI module

Bit 15 — Not Implemented

PCS3–PCS1 — Peripheral Chip Selects 3–1

PCS0/$\overline{SS}$ — Peripheral Chip Select 0/Slave Select

Bit 10 — Not Implemented

MOSI — Master Out Slave In

MISO — Master In Slave Out

These bits determine whether the associated QSM port pin functions as a general-purpose I/O pin or is assigned to the QSPI submodule.

QDDR — QSM Data Direction Register **$YFFC17**

15							7	6	5	4	3	2	1	0
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX							TXD	PCS3	PCS2	PCS1	PCS0/$\overline{SS}$	SCK	MOSI	MISO
RESET:														
							0	0	0	0	0	0	0	0

0 = Input
1 = Output

TXD — Transmit Data
PCS1 — Peripheral Chip Selects 3–1
PSC0/$\overline{SS}$ — Peripheral Chip Select 0/Slave Select
SCK — Serial Clock
MOSI — Master Out Slave In
MISO — Master In Slave Out

SPCR0 — QSPI Control Register 0 **$YFFC18**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MSTR	WOMQ	BITS				CPOL	CPHA	SPBR							
RESET:															
0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0

MSTR — Master/Slave Mode Select
1 = QSPI is system master and can initiate transmission to external SPI devices.
0 = QSPI is a slave device, and only responds to externally generated serial MSTR.

WOMQ — Wired-OR Mode for QSPI Pins
1 = All QSPI port pins designated as output by QDDR function as open-drain outputs.
0 = Output pins have normal outputs instead of open-drain outputs.

BITS — Bits Per Transfer

In master mode, BITS determines the number of data bits transferred for each serial transfer in the queue.

Bit 13–10	Bits per Transfer
0000	16
0001	Reserved
0010	Reserved
0011	Reserved
0100	Reserved
0101	Reserved
0110	Reserved
0111	Reserved
1000	8
1001	9
1010	10
1011	11
1100	12
1101	13
1110	14
1111	15

CPOL — Clock Polarity

1 = The inactive state value of SCK is high.
0 = The inactive state value of SCK is low.

CPHA — Clock Phase

1 = Data is changed on the leading edge of SCK and captured on the following edge of SCK.
0 = Data is captured on the leading edge of SCK and changed on the following edge of SCK.

SPBR — Serial Clock Baud Rate

The QSPI internally generates the baud rate for SCK, the frequency of which is programmable by the user. The following equation determines the SCK baud rate:

$$\text{SCK Baud Rate} = \text{System Clock}/(2 \bullet \text{SPBR})$$

or

$$\text{SPBR} = \text{System Clock}/(2 \bullet \text{SCK Baud Rate Desired})$$

where SPBR equals {2, 3, 4, . . . , 255}.

SPCR — QSPI Control Register **$YFFC1A**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPE	DSCKL							DTL							
RESET:															
0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0

SPE — QSPI Enable

1 = The QSPI is enabled and the pins allocated by QSM register QPAR are controlled by the QSPI.

0 = The QSPI is disabled, and the seven QSPI pins can be used as general-purpose I/O pins, regardless of the values in QPAR.

DSCKL — Delay before SCK

This bit determines the length of time the QSPI delays from peripheral chip select (PCS) valid to SCK transition for serial transfers in which the command control bit, DSCK of the QSPI RAM, equals one.

PCS to SCK Delay = [DSCKL/System Clock Frequency]

where DSCKL equals {1,2,3, . . . 127}.

DTL — Length of Delay after Transfer

These bits determine the length of time that the QSPI delays after each serial transfer in which the command control bit, DT of the QSPI RAM, equals one.

Delay after Transfer = [(32 • DTL)/System Clock Frequency]

where DTL equals {1,2,3, . . . 255}.

SPCR2 — QSPI Control Register 2 **$YFFC1C**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPIFIE	WREN	WRTO	0	ENDQP				0	0	0	0	NEWQP			
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SPIFIE — SPI Finished Interrupt Enable

1 = QSPI interrupts enabled

0 = QSPI interrupts disabled

WREN — Wrap Enable

1 = Wraparound mode enabled

0 = Wraparound mode disabled

WRTO — Wrap To

1 = Wrap to address found in NEWQP

0 = Wrap to address $0

Bit 12 — Not Implemented

ENDQP — Ending Queue Pointer

This field determines the last absolute address in the queue to be completed by the QSPI.

Bits 7–4 — Not Implemented

NEWQP — New Queue Pointer Value
NEWQP determines which queue entry the QSPI transfers first.

SPCR3 — QSPI Control Register **$YFFC1E**

15	14	13	12	11	10	9	8	0
0	0	0	0	0	LOOPQ	HMIE	HALT	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
RESET:								
0	0	0	0	0	0	0	0	

Bits 15–11 — Not Implemented

LOOPQ — QSPI Loop Mode
1 = Feedback path enabled
0 = Feedback path disabled

HMIE — HALTA and MODF Interrupt Enable
1 = HALTA and MODF interrupts enabled
0 = HALTA and MODF interrupts disabled

HALT — Halt
1 = Halt enabled
0 = Halt not enabled
This bit is used by the CPU to stop the QSPI on a queue boundary.

SPSR — QSPI Status Register **$YFFC1F**

15	7	6	5	4	3	2	1	0
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	SPIF	MODF	HALTA	0	CPTQP			
RESET:								
	0	0	0	0	0	0	0	0

SPIF — QSPI Finished Flag
1 = QSPI finished
0 = QSPI not finished
SPIF is set when the QSPI finishes executing the last command determined by the address contained in ENDQP in SPCR2.

MODF — Mode Fault Flag
1 = Another SPI node requested to become the network SPI master while the QSPI was enabled in master mode (MSTR =1), or the PCS0/SS pin was incorrectly pulled low by external hardware.
0 = Normal operation

HALTA — Halt Acknowledge Flag
1 = QSPI halted
0 = QSPI not halted
HALTA is asserted by the QSPI when it has come to an orderly halt at the request of the CPU, through the assertion of HALT.

Bit 4 — Not Implemented

CPTQP — Completed Queue Pointer

CPTQP contains the queue pointer value of the last command in the queue that was completed.

COMD.RAM — Command Ram **$YFFD40**

7	6	5	4	3	2	1	0
CONT	BITSE	DT	DSCK	PCS3	PCS2	PCS1	PCS0*
—	—	—	—	—	—	—	—
CONT	BITSE	DT	DSCK	PCS3	PCS2	PCS1	PCS0*

$YFFD4F

COMMAND CONTROL | PERIPHERAL CHIP SELECT

*The PCS0 bit represents the dual-function PCS0/$\overline{SS}$.

CONT — Continue

1 = Keeps peripheral chip selects asserted after transfer is complete.
0 = Returns control of peripheral chip selects to QPDR after transfer is complete.

BITSE — Bits Enable

1 = Number of bits to transfer defined in BITS field of SPCR0
0 = 8 bits to transfer

DT — Delay after Transfer

1 = Delay
0 = No delay

DSCK — PCS to SCK Delay

1 = DSCKL field in SPCR1 specifies value of delay from PCS valid to SCK
0 = PCS valid to SCK transition is one-half SCK

PCS3–PCS0/$\overline{SS}$ — Peripheral Chip Select

The four peripheral chip-select bits can be used directly to select one of four external chips for the serial transfer, or decoded by external hardware to select one of 16 chip-select patterns for a serial transfer.

Standby Ram (with TPU Emulation)

This module contains 2K bytes of fast static RAM powered by V_{DD} in normal operation. During power-down, the RAM contents are maintained by power on the standby voltage pin, V_{STBY}. The RAM is especially useful for the CPU as system stack space or as variable storage, because of its fast two-cycle access time. Alternately, it can be used by the TPU as emulation RAM for new timer algorithms. Data can be read or written by byte, word, or long-word length. The RAM can be mapped to any 2K-byte boundary in the address map.

RAM Module Programmer's Model

Control Registers

A23					A0	15 8 7 0	
Y	1111	1111	1011	1100	000X	MODULE CONFIGURATION REGISTER (RAMMCR)	$YFFB00
Y	1111	1111	1011	1100	001X	TEST REGISTER (RAMTST)	$YFFB02
Y	1111	1111	1011	1100	010X	BASE ADDRESS/STATUS REGISTER (RAMBAR) 0 0 RAMDS	$YFFB04
Y	1111	1111	1011	1100	011X	NOT IMPLEMENTED	
						• • •	
						• • •	
						• • •	
Y	1111	1111	1111	1100	111X	NOT IMPLEMENTED	

Ram Array

A23	A11		A0	15 8	7 0
*——	—*000	0000	000X		
*——	—*000	0000	001X		
*——	—*000	0000	010X		
*——	—*000	0000	011X		
				• • •	
				• • •	
*——	—*000	0000	111X		

NOTE: All address locations are given in binary.
* = Base address defined in array base address register
X = Zero or one depending on byte/word address
Y = m111 where m is the modmap bit in the SIM MCR (Y = $7 or $F)

RAMMCR — RAM Module Configuration Register **$YFFB00**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STOP	NOT USED						RASP	NOT USED							
RESET:															
0	U						1	U							

STOP — Stop Control
0 = 2K-byte RAM array operates normally.
1 = 2K-byte RAM array enters low-power stop mode.

RASP — RAM Array Space
0 = 2K-byte RAM array is placed in unrestricted space.
1 = 2K-byte RAM array is placed in supervisor space.

RAMTST — RAM Test Register **$YFFBO2**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDTEST							RTBA	NOT USED							
RESET:															
0							0	0							

SDTEST — Soft Defect Test Function
SDTEST is used to select a particular RAM array test, when the system is in test mode. Initiate tests by writing a value to SDTEST. The result is either output on data bit zero (D0) of the intermodule bus (IMB) or is determined by checking the array itself.

RTBA — Base Address Register R/W
0 = RAMBAR may only be written once.
1 = RAMBAR may be written as desired.
RTBA controls the "write-once" lock on the array base address and status register (RAMBAR).

RAMBAR — RAM Base Address and Status Register **$YFFB04**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	NOT USED		RAMDS
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bits 15–3—RAM Array Base Address
These bits specify address lines A23–A11 of the base address of the RAM array when enabled.

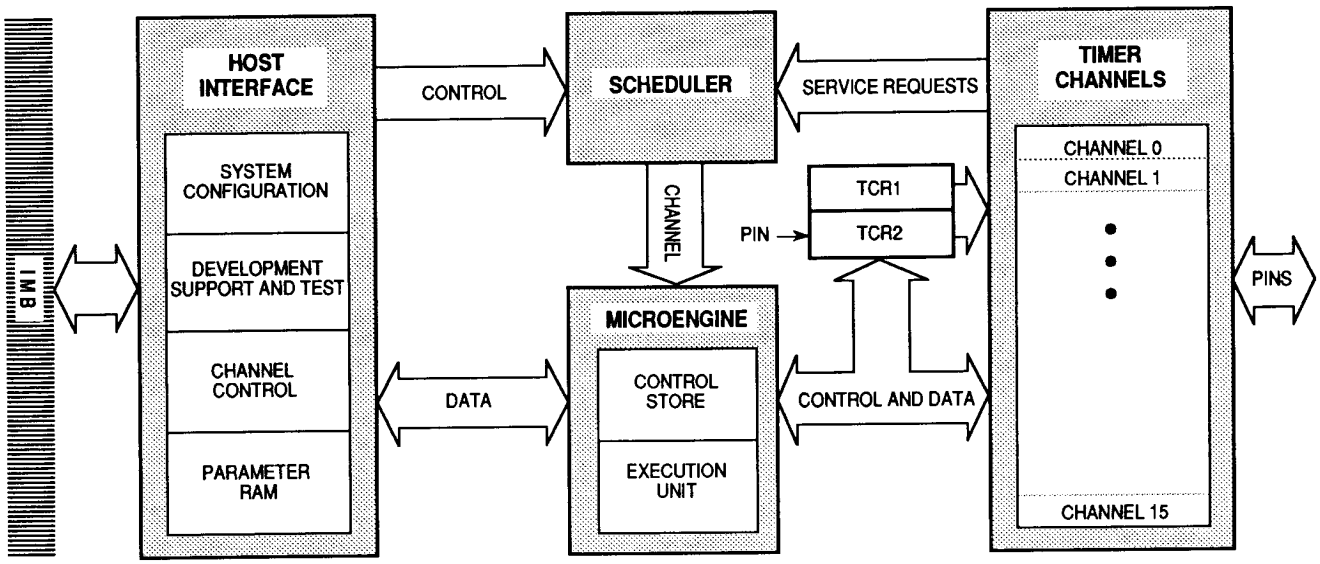
RAMDS — RAM Array Disabled
0 = RAM array is enabled.
1 = RAM array is disabled.

Time Processor Unit (TPU)

The TPU provides optimum performance in controlling time-related activity. The TPU drastically reduces the need for CPU intervention with its dedicated execution unit, tri-level prioritized scheduler, data storage RAM, dual-time bases, and microcode ROM. The TPU controls 16 independent, orthogonal channels, each with associated I/O pin, and is capable of performing any time function. Each channel also contains a dedicated event register, allowing both match and input capture functions.

Features

- 16 Channels — each channel is associated with a pin.
- Each channel can perform any time function.
- Each time function can be assigned to more than one channel at a given time.
- Each channel has an event register comprised of the following:
  - — 16-bit capture register
  - — 16-bit compare/match register
  - — 16-bit greater-than or equal-to comparator
- Each channel can be synchronized to one or both of the two 16-bit free-running timer count registers, TCR1 and TCR2.
- TCR1 is clocked from the output of a programmable prescaler.
- TCR2 is clocked from the output of a prescaler which is also programmable. The prescaler's input is the external T2CLK pin.
- TCR2 can be used as a hardware pulse accumulator, clocked from the external TCR2 pin, or as a gated pulse accumulator of the clock that increments TCR1.
- All channels have at least six 16-bit parameter registers. All parameter registers are contained in a dual-port RAM, accessible from both the TPU and CPU.
- A scheduler has three priority levels.
- All time functions are microcoded.
- Emulation and development support are provided for all time function features such as breakpoint, freeze, and single step, giving internal register accessibility.
- Coherent transfer capability for two parameters is provided in hardware.
- Coherent transfer capability for N parameters can be performed as a TPU microcode function.



TPU Simplified Block Diagram

The following information describes the time functions currently implemented in the TPU microcode ROM.

Discrete Input/Output (DIO)

When a pin is used as a discrete input, a parameter indicates the current input level and the previous 15 levels of a pin. Bit 15, the most significant bit of the parameter, indicates the most recent state. Bit 14 indicates the next most recent state, and so on.

Input Capture/Input Transition Counter (ITC)

Any channel of the TPU can capture the value of a specified TCR upon the occurrence of each transition, and then generate an interrupt request to notify the bus master.

Output Compare (OC)

The time function generates a rising edge, falling edge, or a toggle of the previous edge in one of two ways:

a. OC generates the specified edge at a programmable delay time for a user-specified time. The CPU can also force an immediate output, thereby generating a pulse with a length of the programmable delay time.

b. Upon receiving a link from a channel, OC references, without CPU interaction, a specifiable period or parameter of another channel and adds an offset to that parameter.

Pulse-Width Modulation (PWM)

The TPU can generate a PWM waveform with any duty cycle from zero to 100% (within the resolution and latency capability of the TPU). To define the PWM, the CPU provides one parameter that indicates the period and another parameter that indicates the high time. Updates to one or both of these parameters can direct the waveform change to take effect immediately or coherently, beginning at the next low-to-high transition of the pin.

Synchronized Pulse-Width Modulation (SPWM)

The TPU generates a pulse-width modulated (PWM) waveform in which the CPU can change the period and/or high time at any time. When synchronized to a time function on a second channel, the SPWM low-to-high transitions have a time relationship to transitions on the second channel.

Period Measurement with Additional Transition Detect (PMA)

PMA allows for a special-purpose 16-bit period measurement. It detects the occurrence of an additional transition as indicated by the current period measurement being less than a programmable ratio of the previous period measurement. Once detected, this condition can be counted and compared to a programmable number of additional transitions.

Period Measurement with Missing Transition Detect (PMM)

PMM allows for a special-purpose 16-bit period measurement. It detects the occurrence of a missing transition as indicated by the current period measurement being more than a programmable ratio times the previous period measurement. Once detected, this condition can be counted and compared to a programmable number of transitions.

Position-Synchronized Pulse Generator (PSP)

Any channel of the TPU can generate an output transition or pulse, which is a projection in time based on a reference period previously calculated on another channel.

Stepper Motor (SM)

The stepper motor control algorithm uses a programmable number of step rates to control the linear acceleration and deceleration of a stepper motor.

Any group of up to eight channels can be programmed to generate the control logic necessary to drive a stepper motor. Nominally, only two or four channels are used for a two-phase motor.

Period/Pulse-Width Accumulator (PPWA)

The PPWA algorithm continuously accumulates the high time or the total elapsed interval of a waveform over a programmable number of input periods. It continually tracks the current as well as the most recent accumulated times.

Register Map

	ADDRESS	15 BYTE n 8 | 7 BYTE n + 1 0
S	$YFFE00	MODULE CONFIGURATION REGISTER
S	$YFFE02	CONFIGURATION REGISTER
S	$YFFE04	DEVELOPMENT SUPPORT CONTROL REGISTER
S	$YFFE06	DEVELOPMENT SUPPORT STATUS REGISTER
S	$YFFE08	INTERRUPT CONFIGURATION REGISTER
S	$YFFE0A	INTERRUPT ENABLE REGISTER
S	$YFFE0C	CHANNEL FUNCTION SELECT REGISTER 0
S	$YFFE0E	CHANNEL FUNCTION SELECT REGISTER 1
S	$YFFE10	CHANNEL FUNCTION SELECT REGISTER 2
S	$YFFE12	CHANNEL FUNCTION SELECT REGISTER 3
X	$YFFE14	HOST SEQUENCE REGISTER 0
X	$YFFE16	HOST SEQUENCE REGISTER 1
X	$YFFE18	HOST SERVICE REQUEST REGISTER 0
X	$YFFE1A	HOST SERVICE REQUEST REGISTER 1
S	$YFFE1C	CHANNEL PRIORITY REGISTER 0
S	$YFFE1E	CHANNEL PRIORITY REGISTER 1
S	$YFFE20	INTERRUPT STATUS REGISTER
S	$YFFE22	LINK REGISTER
S	$YFFE24	SERVICE GRANT LATCH REGISTER
S	$YFFE26	DECODED CHANNEL NUMBER REGISTER

S = Supervisor accessible only
X = Assignable as supervisor-accessible only (if SUPV = 1) or unrestricted (if SUPV = 0). Unrestricted allows both user and supervisor access.
Y = m111, where m is the state of the modmap bit in the module configuration register of the system integration module (Y = $7 or $F).

Parameter RAM Map

Channel	Parameter							
	0	1	2	3	4	5	6	7
0	X $YFFF00	02	04	06	08	0A	—	—
1	X $YFFF10	12	14	16	18	1A	—	—
2	X $YFFF20	22	24	26	28	2A	—	—
3	X $YFFF30	32	34	36	38	3A	—	—
4	X $YFFF40	42	44	46	48	4A	—	—
5	X $YFFF50	52	54	56	58	5A	—	—
6	X $YFFF60	62	64	66	68	6A	—	—
7	X $YFFF70	72	74	76	78	7A	—	—
8	X $YFFF80	82	84	86	88	8A	—	—
9	X $YFFF90	92	94	96	98	9A	—	—
10	X $YFFFA0	A2	A4	A6	A8	AA	—	—
11	X $YFFFB0	B2	B4	B6	B8	BA	—	—
12	X $YFFFC0	C2	C4	C6	C8	CA	—	—
13	X $YFFFD0	D2	D4	D6	D8	DA	—	—
14	X $YFFFE0	E2	E4	E6	E8	EA	EC	EE
15	X $YFFFF0	F2	F4	F6	F8	FA	FC	FE

— = Not Implemented
X = Assignable as supervisor accessible only (if SUPV = 1) or unrestricted (if SUPV = 0). Unrestricted allows both user and supervisor access.
Y = m111, where m is the modmap bit in the module configuration register of the system integration module (Y = $7 or $F).

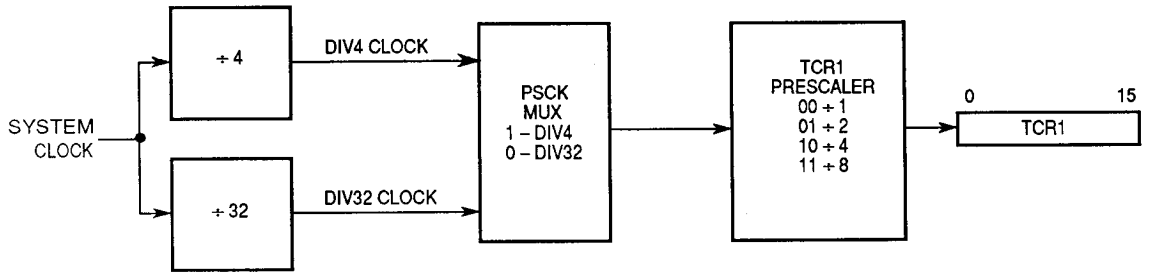
TPU Registers

TMCR — TPU Module Configuration Register **$YFFE00**

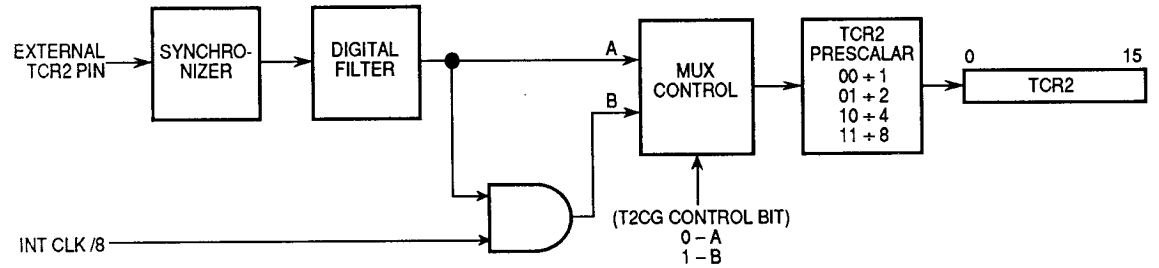
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STOP	TCR1P		TCR2P		EMU	T2CG	STF	SUPV	PSCK	0	0	IARB			
RESET:															
0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0

STOP — Stop Bit
0 = Internal clocks not shut down (reset condition)
1 = Internal clocks shut down

TCR1P — TCR1 Prescaler Control



TCR2P — TCR2 Prescaler Control



EMU — Emulation Control

0 = TPU and RAM not in emulation mode (reset condition)
1 = TPU and RAM in emulation mode

T2CG — TCR2 Clock/Gate Control

0 = TCR2 pin used as clock source for TCR2 (reset condition)
1 = TCR2 pin used as gate of DIV8 clock for TCR2

STF — Stop Flag

0 = TPU operating (reset condition)
1 = TPU stopped

SUPV — Supervisor Data Space

0 = Assignable registers are unrestricted (FC2 is ignored).
1 = (Reset condition) Assignable registers are restricted (FC2 is decoded).

PSCK — Prescaler Clock

0 = DIV32 (system clock/32) is input to TCR1 prescaler.
1 = DIV4 (system clock/4) is input to TCR1 prescaler.

IARB — Interrupt Arbitration ID Bits

This field contains the arbitration number of the TPU that is used to arbitrate for the intermodule bus (IMB) when two or more modules or peripherals have an interrupt on the same priority level.

TCR — Test Configuration Register **$YFFE02**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	INCAD	TCR1C	ACIITR1	ACITR0	0	0	SOSEL2	SOSEL1	SOSEL0	SISEL2	SISEL1	SISEL0	TMM
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits 15–13, 8, 7 — Not Implemented

INCAD — Increment Address

When set, INCAD forces the μPC to increment once for each assertion of the ACUTL line. This feature is used to sequentially dump the μROM.

0 = Normal operation
1 = μPC increments

TCR1C–TCR1 Clock

TCR1C selects the clock source, internal or external, for TCR1. This bit can be read or written in test mode only.

0 = Normal operation (TCR1 clocked internally)
1 = TCR1 clocked externally via the TCR2 pin; TCR1 is now clocked in parallel with TCR2.

ACUTR1–0 — Activate Circuit Under Test Response 1, 0

The ACUTR bits determine the TPU module response to the test submodule asserting ACUT only.

ACUTR1–0	TPU Response
00	None
01	Run One TPU Microcycle (Single Step)
10	Assert Scheduler End-of-Time Slot (Test HS PLA)
11	Reserved

SOSEL2–0 — Scan-Out Select 2–0

These bits define the TPU output scan path to be connected to master shift register B (MSRB) through the SCANB line.

SOSEL1–0	Output Scan Path
000	None
001	μPC
010	Microinstruction
011	Branch PLA
100	μPC Breakpoint
101	Scheduler PLA
110	Channel Breakpoint
111	Reserved

SISEL2–SISEL0 — Scan-In Select 2–0

The bits define the TPU input scan path to be connected to master shift register A (MSRA) through the SCANA line.

SISEL1–0	Input Scan Path
000	None
001	μPC
010	Microinstruction
011	Branch PLA
100	μPC Breakpoint
101	Scheduler PLA
110	Channel Breakpoint
111	Reserved

TMM—Test Memory Map

0 = Normal memory map

1 = Module located at $YDF000–$YDFFFF

DSCR — Development Support Control Register **$YFFE04**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HOT4	0	0	0	0	BLC	CLKS	FRZ1	FRZ0	CCL	BP	BC	BH	BL	BM	BT
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

HOT4 — Hang on T4

0 = Exit wait on T4 state caused by assertion of HOT4

1 = Enter wait on T4 state

Bits 14–11 — Not Implemented

BLC — Branch Latch Control

1 = Do not latch conditions into branch condition register before exiting the halted state or during the time-slot transition period.

0 = Latch conditions into branch condition register prior to exiting halted state.

CLKS — Stop Clocks (to TCRs)

0 = Do not stop TCRs.

1 = Stop TCRs during the halted state.

FRZ1–0 — IMB FREEZE Response

The FRZ bits specify the TPU microengine response to the FREEZE signal.

FRZ1–0	TPU Response
00	Ignore Freeze
01	Reserved
10	Freeze at End of Current Microcycle
11	Freeze at Next Time-Slot Boundary

CCL — Channel Conditions Latch

CCL controls the latching of channel conditions (MRL and TDL) when the CHAN register is written.

0 = Only the pin state condition of the new channel is latched as a result of the write CHAN register microinstruction.

1 = Pin state, MRL, and TDL conditions of the new channel are latched as a result of a write CHAN register microinstruction.

BP, BC, BH, BL, BM, and BT — Breakpoint Enable Bits

Bits 5–0 contain the various breakpoint enable bits for the TPU, specifying the conditions for a breakpoint. The breakpoint is enabled by setting the corresponding bit to one and is disabled by setting the bit to zero.

BP — Break if µPC equals µPC breakpoint register.

BC — Break if CHAN register equals channel breakpoint register at beginning of state or when CHAN is changed through microcode.

BH — Break if host service latch is asserted at beginning of state.

BL — Break if link service latch is asserted at beginning of state.

BM — Break if MRL is asserted at beginning of state.

BT — Break if TDL is asserted at beginning of state.

DSSR — Development Support Status Register **$YFFE06**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	BKPT	PCBK	CHBK	SRBK	TPUF	0	0	0
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits 15–8, 2–0 — Not Implemented

BKPT — Breakpoint Asserted Flag

If an internal breakpoint caused the TPU to enter the halted state, the TPU asserts the BKPT signal on the IMB and the BKPT flag. The TPU continues to assert BKPT until it recognizes a breakpoint acknowledge cycle from a host, or until the FREEZE signal on the IMB is asserted.

PCBK — µPC Breakpoint Flag

PCBK is asserted if a breakpoint occurs because of a µPC register match with the µPC breakpoint register. PCBK is negated when the BKPT flag is negated.

CHBK — Channel Register Breakpoint Flag

CHBK is asserted if a breakpoint occurs because of a CHAN register match with the channel register breakpoint register. CHBK is negated when the BKPT flag is negated.

SRBK — Service Request Breakpoint Flag

SRBK is asserted if a breakpoint occurs because of any of the service request latches being asserted along with their corresponding enable flag in the development support control register. SRBK is negated when the BKPT flag is negated.

TPUF — TPU FREEZE Flag

TPUF is asserted whenever the TPU is in a halted state as a result of FREEZE being asserted. This flag is automatically negated when the TPU exits the halted state because of FREEZE being negated.

TICR — TPU Interrupt Configuration Register **$YFFE08**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	CIRL			CIBV				0	0	0	0
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits 15–11 — Not Implemented

CIRL — Channel Interrupt Request Level

The interrupt request level for all channels is specified by this three-bit encoded field. Level seven for this field indicates a nonmaskable interrupt; level zero indicates that all channel interrupts are disabled.

CIBV — Channel Interrupt Base Vector

This field specifies the most significant nibble of all 16 TPU channel interrupt vector numbers.

Bits 3–0 — Not Implemented

CIER — Channel Interrupt Enable Register **$YFFE0A**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH 15	CH 14	CH 13	CH 12	CH 11	CH 10	CH 9	CH 8	CH 7	CH 6	CH 5	CH 4	CH 3	CH 2	CH 1	CH 0
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CH 15–0 — Interrupt Enable/Disable for each Channel

0 = Channel interrupts disabled

1 = Channel interrupts enabled

CFSR0–3 — Channel Function Select Register 0–3 **$YFFE0C–$YFFE12**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH (15) (11) (7) (3)				CH (14) (10) (6) (2)				CH(3) (9) (5) (1)				CH(12) (8) (4) (0)			
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits15–0 — Encoded One of 16 Time Functions for each Channel

HSQR0 — Host Sequence Register 0 **$YFFE14**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH 15		CH 14		CH 13		CH 12		CH 11		CH 10		CH 9		CH 8	
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

HSQR1 — Host Sequence Register 1 **$YFFE16**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH 7		CH 6		CH 5		CH 4		CH 3		CH 2		CH 1		CH 0	
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CH 15–0 — Encoded Host Sequence

HSRR0 — Host Service Request Register 0 **$YFFE18**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH 15		CH 14		CH 13		CH 12		CH 11		CH 10		CH 9		CH 8	
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

HSRR1 — Host Service Request Register 1 **$YFFE1A**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH 7		CH 6		CH 5		CH 4		CH 3		CH 2		CH 1		CH 0	
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CH 15–0 — Encoded Type of Host Service

Bit 1	Bit 0	Service
0	0	No Host Service (Reset Condition)
0	1	Type 1 Host Service
1	0	Type 2 Host Service
1	1	Type 3 Host Service

Function Name	Function Code	Host Service Request Code	Host Sequence Code*
DIO Discrete Input/Output	$8	1= Force Output High	0 = Trans Mode — Record Pin on Transition
		2 = Force Output Low	0 = Trans Mode — Record Pin on Transition
		3 = Initialization, Input Specified	0 = Trans Mode — Record Pin on Transition
		3 = Initialization, Periodic Input	1 = Match Mode — Record Pin at Match_Rate
		3 = Update Pin Status Parameter	2 = Record Pin State on HSR 11
ITC Input Capture/ Input Transition Counter	$A	0 = None 1 = Initialization 2 = (Not Implemented) 3 = (Not Implemented)	0 = No Link, Single Mode 1 = No Link, Continuous Mode 2 = Link, Single Mode 3 = Link, Continuous Mode
OC Output Compare	$E	0 = None 1 = Host-Initiated Pulse Mode 2 = (Not Implemented) 3 = Continuous Pulse Mode	0 = Execute All Functions 1 = Execute All Functions 2 = Only Update TCRn Parameters 3 = Only Update TCRn Parameters
PWM Pulse-Width Modulation	$9	0 = None 1 = Immediate Update Request 2 = Initialization 3 = (Not Implemented)	(None Implemented)
SPWM Synchronized Pulse-Width Modulation	$7	0 = None 1 = (Not Implemented) 2 = Initialization 3 = Immediate Update Request	0 = Mode 0 1 = Mode 1 2 = Mode 2 3 =(Not Implemented)
PMA/PMM Period Measurement with Additional/Missing Transition Detect	$B	0 = None 1 = Initialization 2 = (Not Implemented) 3 = (Not Implemented)	0 = PMA Bank Mode 1 = PMA Count Mode 2 = PMM Bank Mode 3 = PMM Count Mode
PSP Position-Synchronized Pulse Generator	$C	0 = None 1 = Immediate Update Request 2 = Initialization 3 = Force Change	0 = Pulse Width Set by Angle 1 = Pulse Width Set by Time 2 = Pulse Width Set by Angle 3 = Pulse Width Set by Time
SM Stepper Motor	$D	0 = None 1 = None 2 = Initialization 3 = Step Request	(None Implemented)
PPWA Period/Pulse-Width Accumulator	$F	0 = None 1 = (Not Implemented) 2 = Initialization 3 = (Not Implemented)	0 = 24-Bit Period 1 = 16-Bit Period + Link 2 = 24-Bit Pulse Width 3 = 16-Bit Pulse Width + Link

* Host Sequence Code interpretation is determined by the function; some HSQ codes apply to all HSR codes, some to only one, such as Init.

CPR0 — Channel Priority Register 0 **$YFFE1C**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH 15		CH 14		CH13		CH 12		CH 11		CH 10		CH 9		CH 8	
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CPR1 — Channel Priority Register 1 **$YFFE1E**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH 7		CH 6		CH 5		CH 4		CH 3		CH 2		CH 1		CH 0	
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CH 15–0 — Encoded One of Three Channel Priority Levels

Bit 1	Bit 0	Service
0	0	Channel Disabled (Reset Condition)
0	1	Low
1	0	Middle
1	1	High

CISR — Channel Interrupt Status Register **$YFFE20**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH 15	CH 14	CH 13	CH 12	CH 11	CH 10	CH 9	CH 8	CH 7	CH 6	CH 5	CH 4	CH 3	CH 2	CH 1	CH 0
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CH 15–0 — Interrupt Status Bit
0 = Channel interrupt not asserted (reset condition)
1 = Channel interrupt asserted

LR — Link Register **$YFFE22**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH 15	CH 14	CH 13	CH 12	CH 11	CH 10	CH 9	CH 8	CH 7	CH 6	CH 5	CH 4	CH 3	CH 2	CH 1	CH 0
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CH 15–0 — Test Mode Link Service Request Enable Bit
0 = Link bit not asserted
1 = Link bit asserted

SGLR — Service Grant Latch Register **$YFFE24**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH 15	CH 14	CH 13	CH 12	CH 11	CH 10	CH 9	CH 8	CH 7	CH 6	CH 5	CH 4	CH 3	CH 2	CH 1	CH 0
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CH 15–0 — Service Granted Bits

DCNR — Decoded Channel Number Register **$YFFE26**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH 15	CH 14	CH 13	CH 12	CH 11	CH 10	CH 9	CH 8	CH 7	CH 6	CH 5	CH 4	CH 3	CH 2	CH 1	CH 0
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CH 15–0 — Service Status Bits