MC68HC16V1

Technical Summary

16-Bit Modular Microcontroller

1 Introduction

The MC68HC16V1 is a high-speed 16-bit modular microcontroller. It is a member of the M68300/68HC16 family of modular microcontrollers.

M68HC16 controllers are built up from standard modules that interface through a common internal bus. Standardization facilitates rapid development of devices tailored for specific applications.

The MC68HC16V1 incorporates a true 16-bit central processing unit (CPU16), a stream-lined integration module (SLIM), a queued serial module (QSM), and a general-purpose timer (GPT). These modules are interconnected by the intermodule bus (IMB).

Maximum system clock for the MCU is 20.97 MHz. A phase-locked loop circuit synthesizes the clock from a frequency reference. Either a crystal (32.768 kHz for slow reference mode or 4.194 MHz for fast reference mode) or an externally generated signal can be used. System hardware and software support changes in clock rate during operation. Because the MCU is a fully static design, register and memory contents are not affected by clock rate changes.

High-density complementary metal-oxide semiconductor (HCMOS) architecture makes the basic power consumption of the MCU low. Power consumption can be minimized by stopping the system clock. The M68HC16 instruction set includes a low-power stop (LPSTOP) command that efficiently implements this capability.

Table 1 Standard Device Ordering Information

Package Type	Shadow Register Contents	Frequency	Temperature	Package Order Quantity	Order Number
100-Pin TQFP	MCRC: $984F PCON: $15C6	20 MHz	– 40 to + 85 °C	2 pc tray	SPMC16V1CPU20
				84 pc tray	MC68HC16V1CPU20
				420 pc tray	MC16V1CPU20B1

TABLE OF CONTENTS

1.1 Features

- CPU16
  - — 16-bit architecture
  - — Full set of 16-bit instructions
  - — Three 16-bit index registers
  - — Two 16-bit accumulators
  - — 16-bit multiply and accumulate (digital signal processing support)
  - — High-level language support
  - — Fast interrupt response time
  - — Hardware breakpoint signal
  - — Background debugging mode
  - — Fully static implementation
- Stream-Lined Integration Module (SLIM)
  - — Multiplexed/non-multiplexed external bus support
  - — Three programmable chip-select outputs
  - — 512-byte boot ROM
  - — System protection logic, including improved loss of clock protection, watchdog timer and bus monitor
  - — Automatic configuration from shadow registers
  - — On-chip phase-locked loop (PLL) for system clock supports slow and fast reference modes
- Queued Serial Module (QSM)
  - — Enhanced serial communication interface (SCI)
  - — Queued serial peripheral interface (QSPI)
  - — Dual function I/O ports
- General-Purpose Timer (GPT)
  - — 16-bit free-running counter with eight-stage prescaler
  - — Three input capture channels
  - — One input capture/output compare channel
  - — Four output compare channels
  - — One pulse accumulator/event counter input

1.2 Block Diagram

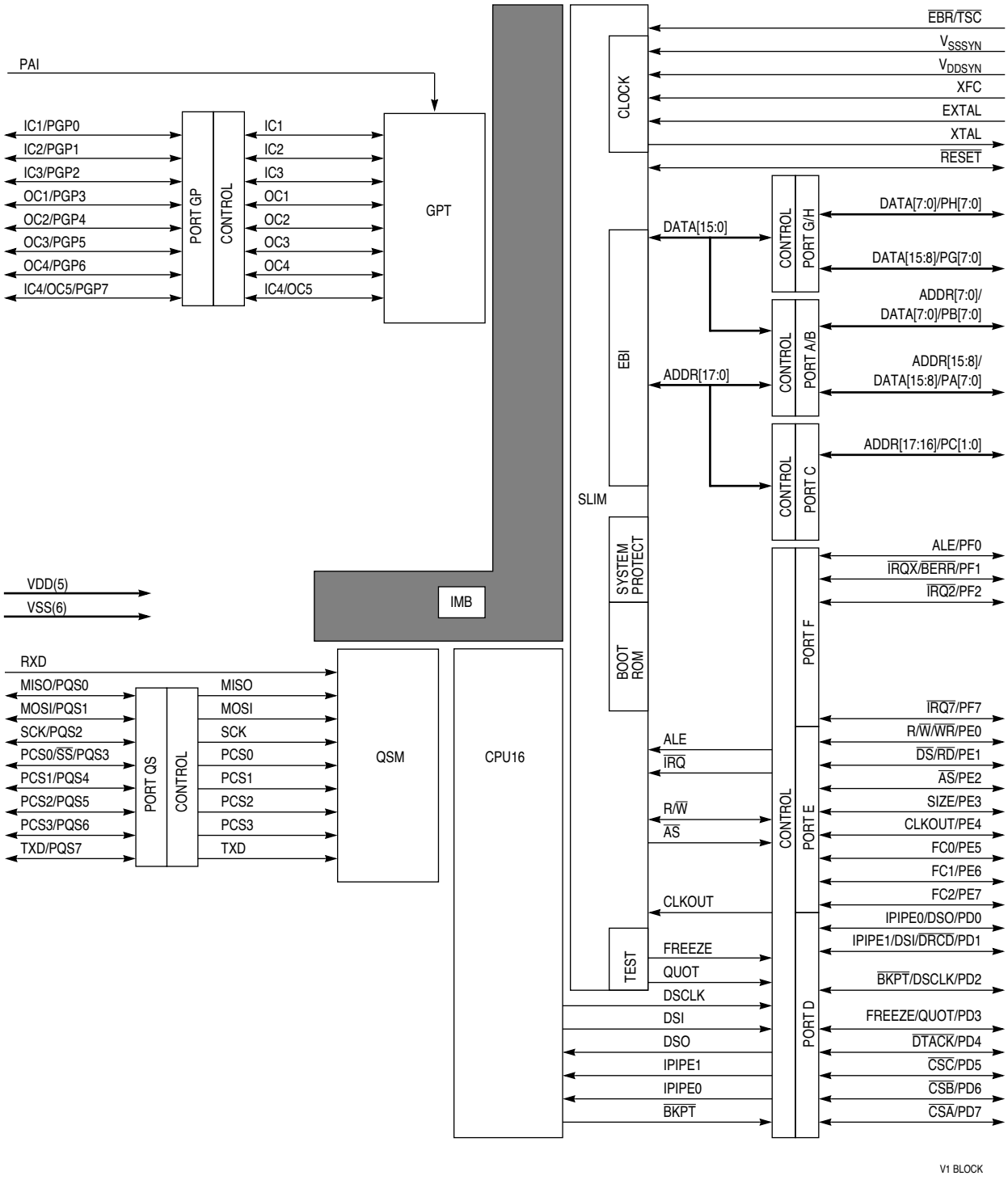


Figure 1 MC68HC16V1 Block Diagram

1.3 Pin Assignments

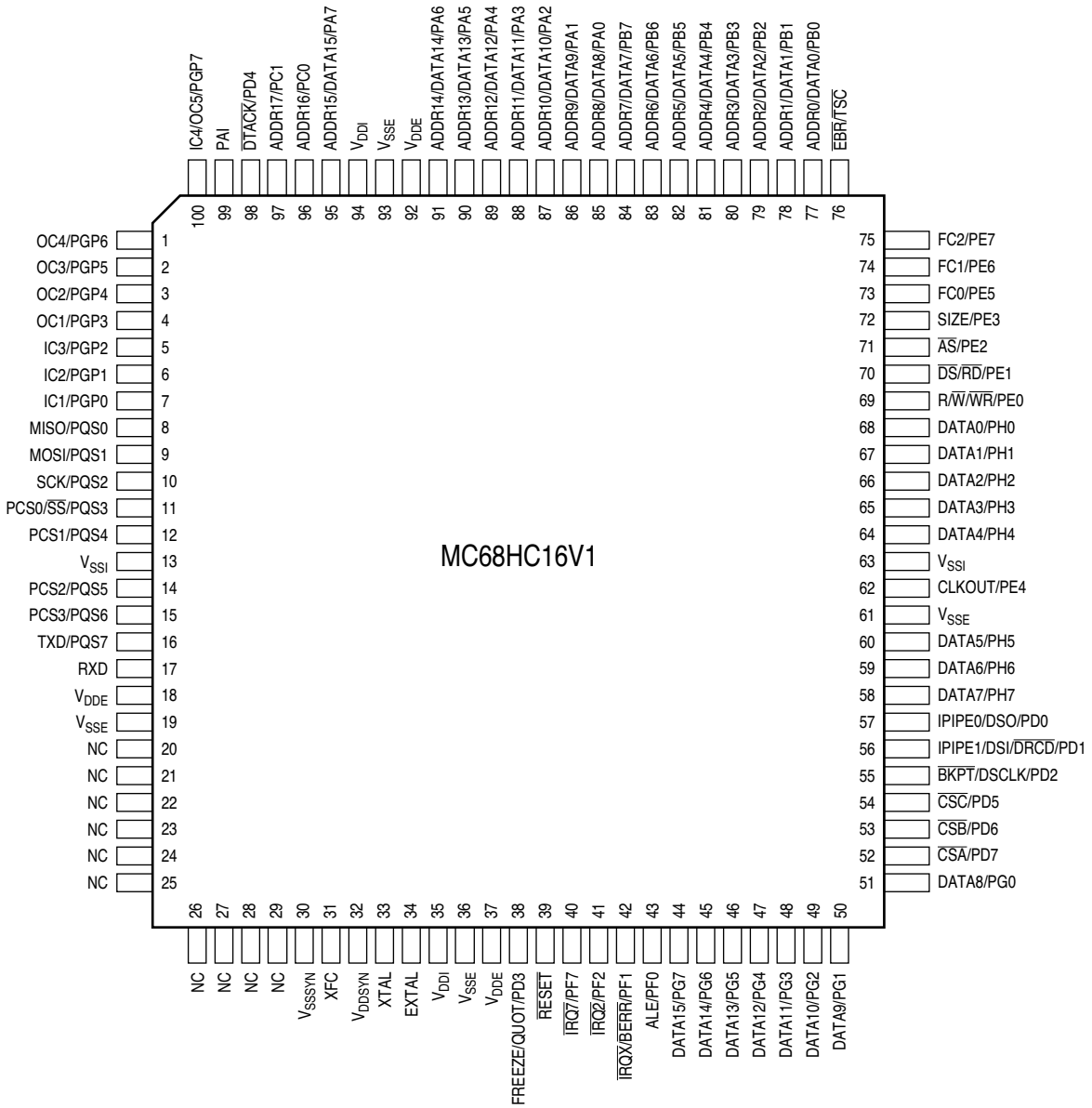


Figure 2 MC68HC16V1 Pin Assignments

1.4 Address Map

Figure 3 is a map of the MCU internal addresses. The boot ROM may or may not be enabled, depending on the system configuration during reset. Unimplemented blocks are mapped externally.

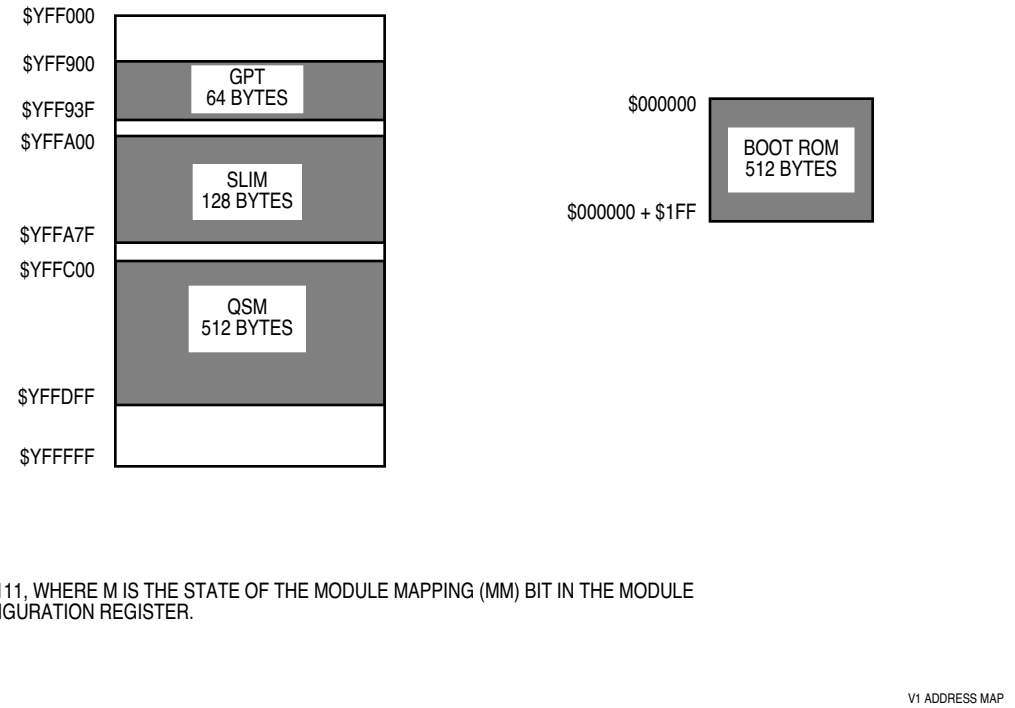


Figure 3 MC68HC16V1 Address Map

1.5 Intermodule Bus

The intermodule bus (IMB) is a standardized bus developed to facilitate design and operation of modular microcontrollers. It contains circuitry that supports exception processing, address space partitioning, multiple interrupt levels, and vectored interrupts. MCU modules communicate with one another and with external components via the IMB. Although the full IMB supports 24 address and 16 data lines, the CPU16 uses only 20 address and 16 data lines. ADDR[23:20] follow the logic state of ADDR19. Addresses $080000 to $F7FFFF are not accessible. Only ADDR[17:0] are externally accessible on the MC68HC16V1.

2 Signal Descriptions

2.1 MC68HC16V1 Pin Characteristics

Table 2 shows MCU pins and their characteristics. All inputs detect CMOS logic levels. All inputs can be put in a high-impedance state, but the method of doing this differs depending upon pin function. Refer to **Table 4** for a description of output drivers. An entry in the discrete I/O column of the MCU pin characteristics table indicates that a pin has an alternate I/O function. The port designation is given when it applies. Refer to **Figure 1** for information about port organization.

Table 2 MCU Pin Characteristics

Pin Mnemonic	Output Driver	Input Synchronized	Input Hysteresis	Discrete I/O	Port Designation
ADDR[17:16]	Aw	—[1]	No	I/O	PC[1:0]
ADDR/DATA[15:8]	Aw	—[1]	No	I/O	PA[7:0]
ADDR/DATA[7:0]	Aw	—[1]	No	I/O	PB[7:0]
ALE	A	—[1]	No	I/O	PF0
$\overline{AS}$	Bw	—[1]	No	I/O	PE2
$\overline{BKPT}$/DSCLK	—	Yes	Yes	I/O	PD2
CLKOUT	A	—[1]	No	I/O	PE4
$\overline{CSA}$	Aw	—[1]	No	I/O	PD7
$\overline{CSB}$	B	—[1]	No	I/O	PD6
$\overline{CSC}$	Aw	—[1]	No	I/O	PD5
DATA[15:8]	A	—[1]	No	I/O	PG[7:0]
DATA[7:0]	A	—[1]	No	I/O	PH[7:0]
$\overline{DS}$/$\overline{RD}$	Bw	—[1]	No	I/O	PE1
$\overline{DTACK}$	B	Yes	No	I/O	PD4
$\overline{EBR}$/$\overline{TSC}$	Bw	Yes[2]	Yes	—	—
EXTAL	—	—	—[3]	—	—
FC[2:0]	Aw	—[1]	No	I/O	PE[7:5]
FREEZE/QUOT	Aw	—[1]	No	I/O	PD3
IC4/OC5	A	Yes	Yes	I/O	PGP7
IC[3:1]	A	Yes	Yes	I/O	PGP[2:0]
IPIPE1/DSI/$\overline{DRCD}$	A	—	—	I/O	PD1
IPIPE0/DSO	A	Yes	Yes	I/O	PD0
$\overline{IRQX}$/$\overline{BERR}$	Bw	Yes	Yes	I/O	PF1
$\overline{IRQ7}$	Bw	Yes	Yes	I/O	PF7
$\overline{IRQ2}$	Bw	Yes	Yes	I/O	PF2
MISO	Bo	Yes[4]	Yes	I/O	PQS0
MOSI	Bo	Yes[4]	Yes	I/O	PQS1
OC[4:1]	A	Yes	Yes	I/O	PGP[6:3]
PAI	—	Yes	Yes	I	—
PCS[3:1]	Bo	Yes[4]	Yes	I/O	PQS[6:4]
PCS0/$\overline{SS}$	Bo	Yes[4]	Yes	I/O	PQS3
R/$\overline{W}$/$\overline{WR}$	Bw	—[1]	No	I/O	PE0

Table 2 MCU Pin Characteristics (Continued)

Pin Mnemonic	Output Driver	Input Synchronized	Input Hysteresis	Discrete I/O	Port Designation
$\overline{\text{RESET}}$	Bo	Yes	Yes	—	—
$\overline{\text{RXD}}$	—	No	No	—	—
SCK	Bo	Yes[4]	Yes	I/O	PQS2
SIZE	Bw	—	No	I/O	PE3
TXD	Bo	Yes[4]	Yes	I/O	PQS7
XFC	—	—	—	—	—
XTAL	—	—	—	—	—

1. Synchronized input used only if pin configured as a digital I/O pin.
2. Unsynchronized input used for external bus request ($\overline{\text{EBR}}$) function.
3. Hysteresis if PLL enabled.
4. DATA[15:8]/PG[7:0] and DATA[7:0]/PH[7:0] synchronized only during reset. QSM pins synchronized only if used as port I/O pins.

2.2 MC68HC16V1 Power Connections

Table 3 MCU Power Connections

Pin	Description
V_{DDSYN}	Clock synthesizer power/Clock mode select
V_{SSSYN}	Clock synthesizer ground
V_{DDE}, V_{SSE}	External periphery power (source and drain)
V_{DDI}, V_{SSI}	Internal module power (source and drain)

2.3 MC68HC16V1 Output Driver Types

Table 4 MCU Output Driver Types

Type	I/O	Description
A	O	Output-only signals that are always driven; no external pull-up required.
Aw	O	Type A output with weak P-channel pull-up during reset.
B	O	Three-state output that includes circuitry to pull up output before high impedance is established, to ensure rapid rise time. An external holding resistor is required to maintain logic level while the pin is in the high-impedance state.
Bo	O	Type B output that can be operated in an open-drain mode.
Bw	O	Type B output with weak P-channel pull-up during reset.

2.4 MC68HC16V1 Signal Characteristics

Table 5 MCU Signal Characteristics

Signal Name	MCU Module	Signal Type	Active State
ADDR[17:0]	SLIM	Bus	—
ALE	SLIM	Output	1
$\overline{AS}$	SLIM	Output	0
$\overline{BKPT}$	CPU16	Input	0
CLKOUT	SLIM	Output	—
$\overline{CSA}$, $\overline{CSB}$, $\overline{CSC}$	SLIM	Output	0
DATA[15:0]	SLIM	Bus	—
$\overline{DRCD}$	SLIM	Input	0
$\overline{DS}$	SLIM	Output	0
DSI	CPU16	Input	0
DSO	CPU16	Output	0
DSCLK	CPU16	Input	Serial Clock
$\overline{DTACK}$	SLIM	Input	0
$\overline{EBR}$/$\overline{TSC}$	SLIM	Input	0
EXTAL	SLIM	Input	—
FC[2:0]	SLIM	Output	—
FREEZE	SLIM	Output	1
IC[3:1]	GPT	Input	—
IPIPE[1:0]	CPU16	Output	—
$\overline{IRQ7}$, $\overline{IRQ2}$	SLIM	Input	0
$\overline{IRQX}$/$\overline{BERR}$	SLIM	Input	0
MISO	QSM	Input/Output	—
MOSI	QSM	Input/Output	—
OC[4:1]	GPT	Output	—
PAI	GPT	Input	—
PCS0/$\overline{SS}$	QSM	Output	—
PCS[3:1]	QSM	Output	—
QUOT	SLIM	Output	—
$\overline{RD}$	SLIM	Output	0
R/$\overline{W}$	SLIM	Output	0
$\overline{RESET}$	SLIM	Input/Output	0
RXD	QSM	Input	—
SCK	QSM	Input/Output	—
SIZE	SLIM	Output	1
TXD	QSM	Output	—
$\overline{WR}$	SLIM	Output	0
XFC	SLIM	Input	—
XTAL	SLIM	Output	—

2.5 MC68HC16V1 Signal Function

Table 6 MCU Signal Function

Signal Name	Mnemonic	Function
Address Bus	ADDR[17:0]	18-bit address used by CPU16
Address Latch Enable	ALE	Enables external address latching by indicating when address is valid on address/data bus
Address Strobe	$\overline{AS}$	Indicates that a valid address is on the address bus
Bus Error	BERR	Signals to the SLIM that a bus error has occurred
Breakpoint	$\overline{BKPT}$	Signals a hardware breakpoint to the CPU
Clock Out	CLKOUT	System clock output
Chip Selects	$\overline{CSA}$, $\overline{CSB}$, $\overline{CSC}$	Select external devices at programmed addresses. $\overline{CSB}$ can address boot memory at reset
Data Bus	DATA[15:0]	16-bit data bus
Data Strobe	$\overline{DS}$	Indicates that an external device should place valid data on the data bus during a read cycle and that valid data has been placed on the bus by the CPU during a write cycle
Data Transfer Acknowledge	$\overline{DTACK}$	Acknowledges to the SLIM that data has been received for a write cycle, or that data is valid on the data bus for a read cycle
Development Serial I/O, Clock	DSI, DSO, DSCLK	Serial I/O for background debug mode and clock for background debug mode
Drive Reset Configuration Data	$\overline{DRCD}$	Causes the MCU to take configuration information from inputs to the SLIM during reset instead of from the MCRC and PCON registers
Crystal Oscillator	EXTAL, XTAL	Connections for clock synthesizer circuit reference; a crystal or an external oscillator can be used
External Bus Request	$\overline{EBR}$	Prevents MCU from being able to use external bus
External Filter Capacitor	XFC	Connection for external phase-locked loop filter capacitor
Freeze	FREEZE	Indicates that the CPU16 has entered background mode
Function Codes	FC[2:0]	Identifies processor state and current address space
Programmable Interrupt Request Level	$\overline{IRQX}$	Provides a programmable interrupt priority level to the CPU; can also be configured as bus error input indication
Input Capture Inputs	IC[3:1]	GPT input capture inputs
Interrupt Request Inputs	$\overline{IRQ7}$, $\overline{IRQ2}$	Request interrupt service from CPU16
Instruction Pipeline	IPIPE[1:0]	Indicates instruction pipeline activity
Master In Slave Out	MISO	Serial input to QSPI in master mode; serial output from QSPI in slave mode
Master Out Slave In	MOSI	Serial output from QSPI in master mode; serial input to QSPI in slave mode
Peripheral Chip-Selects	PCS[3:0]	QSPI peripheral chip selects
Port A	PA[7:0]	Port A digital input or output signals
Port B	PB[7:0]	Port B digital input or output signals
Port C	PC[1:0]	Port C digital input or output signals
Port D	PD[7:0]	Port D digital input or output signals
Port E	PE[7:0]	Port E digital input or output signals
Port F	PF7, PF[2:0]	Port F digital input or output signals
Port G	PG[7:0]	Port G digital input or output signals
Port GP	PGP[7:0]	GPT digital input or output signals
Port H	PH[7:0]	Port H digital input or output signals
Port QS	PQS[7:0]	QSM digital input or output signals

Table 6 MCU Signal Function (Continued)

Signal Name	Mnemonic	Function
Pulse accumulator input	PAI	Input to the GPT pulse accumulator
Quotient Out	QUOT	Provides the quotient bit of the polynomial divider
Read Enable	$\overline{RD}$	Indicates a read operation on the data bus
Read/Write	R/$\overline{W}$	Indicates the direction of data transfer on the bus
Reset	$\overline{RESET}$	System reset
SCI Receive Data	RXD	Serial input to the SCI
QSPI Serial Clock	SCK	Clock output from QSPI in master mode; clock input to QSPI in slave mode
Size	SIZE	Indicates the number of bytes to be transferred during a bus cycle
Slave Select	$\overline{SS}$	Causes serial transmission when QSPI is in slave mode; causes mode fault in master mode
Three-State Control	$\overline{TSC}$	Places all output drivers in a high-impedance state
SCI Transmit Data	TXD	Serial output from the SCI
Write Enable	$\overline{WR}$	Indicates a write operation on the data bus

3 Stream-Lined Integration Module

The MC68HC16V1 stream-lined integration module (SLIM) consists of six functional blocks that control system start-up, initialization, configuration, and the external bus. **Figure 4** shows the SLIM block diagram.

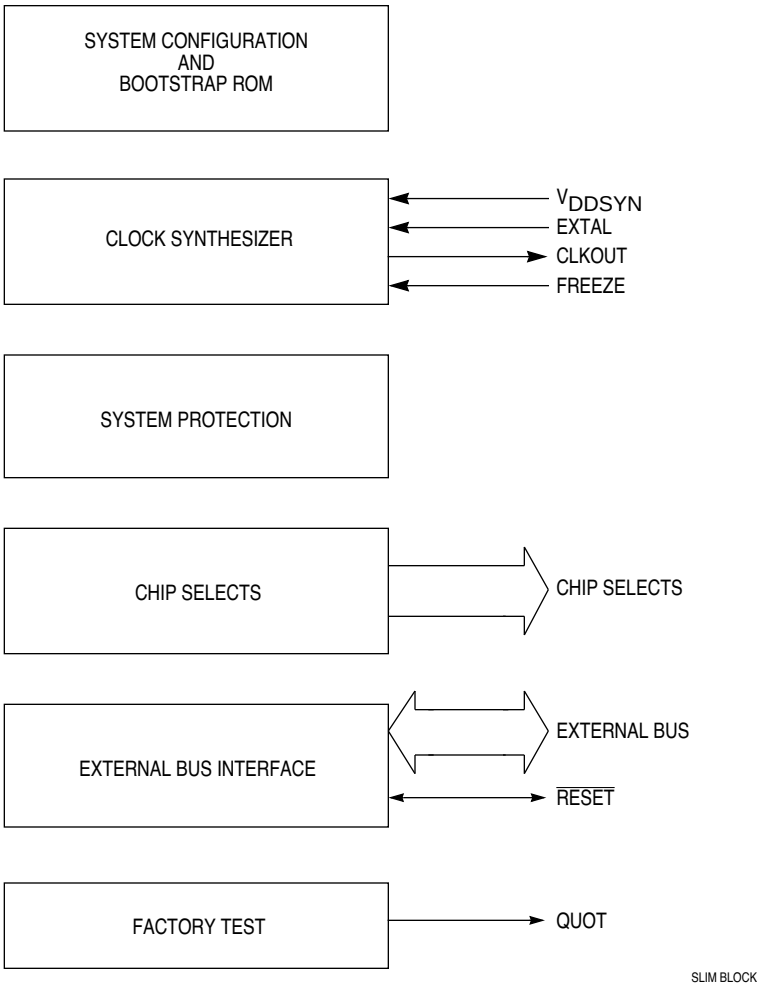


Figure 4 SLIM Block Diagram

3.1 Overview

The system configuration block controls MCU configuration and operating mode. The bootstrap ROM provides the system with a small amount of mask programmable memory used for initialization.

The clock synthesizer generates clock signals used by the SLIM, other IMB modules, and external devices.

The system protection block provides bus monitor functions, software watchdog features, a real-time clock, and a prescaler.

The chip-select block provides three chip-select signals. Each chip-select has an individual base address register and option register.

The external bus interface handles the transfer of information between IMB modules and external address space.

The system test block incorporates hardware necessary for testing the MCU. It is used to perform factory tests, and its use in normal applications is not supported.

Table 7 shows the SLIM address map, which occupies 128 bytes. Unused registers within the 128-byte address space return zeros when read.

Table 7 SLIM Address Map

Address	15 8	7 0
$YFFA00[1]	SLIM MODULE CONFIGURATION REGISTER (SLIMCR)	
$YFFA02	SLIM TEST REGISTER (SLIMTR)	
$YFFA04	CLOCK SYNTHESIZER CONTROL REGISTER (SYNCR)	
$YFFA06	NOT USED	RESET STATUS REGISTER (RSR)
$YFFA08	NOT USED	SLIM TEST REGISTER E (SLIMTRE)
$YFFA0A	MODULE CONFIGURATION SHADOW REGISTER (MCRC)	
$YFFA0C	NOT USED	SUB-MODULE DISABLE (SMD)
$YFFA0E	PORT/CLOCK CONFIGURATION SHADOW REGISTER (PCON)	
$YFFA10	PORT A OUTPUT DATA (PORTA)	PORT B OUTPUT DATA (PORTB)
$YFFA12	PORT A PIN DATA (PORTAP)	PORT B PIN DATA (PORTBP)
$YFFA14	NOT USED	PORT A/B DATA DIRECTION (DDRAB)
$YFFA16	RESERVED	
$YFFA18	PORT C OUTPUT DATA (PORTC)	PORT D OUTPUT DATA (PORTD)
$YFFA1A	PORT C PIN DATA (PORTCP)	PORT D PIN DATA (PORTDP)
$YFFA1C	PORT C DATA DIRECTION (DDRC)	PORT D DATA DIRECTION (DDRD)
$YFFA1E	PORT C PIN ASSIGNMENT (PCPAR)	PORT D PIN ASSIGNMENT (PDPAR)
$YFFA20	NOT USED	PORT E OUTPUT DATA (PORTE)
$YFFA22	NOT USED	PORT E PIN DATA (PORTEP)
$YFFA24	NOT USED	PORT E DATA DIRECTION (DDRE)
$YFFA26	NOT USED	PORT E PIN ASSIGNMENT (PEPAR)
$YFFA28	PORT G OUTPUT DATA (PORTG)	PORT H OUTPUT DATA (PORTH)
$YFFA2A	PORT G PIN DATA (PORTGP)	PORT H PIN DATA (PORTHP)
$YFFA2C	PORT G DATA DIRECTION (DDRG)	PORT H DATA DIRECTION (DDRH)
$YFFA2E	RESERVED	
$YFFA30	NOT USED	PORT F OUTPUT DATA (PORTF)
$YFFA32	NOT USED	PORT F PIN DATA (PORTFP)
$YFFA34	NOT USED	PORT F DATA DIRECTION (DDRF)
$YFFA36	PORT F PIN ASSIGNMENT (PFPAR)	
$YFFA38	NOT USED	PORT F EDGE FLAGS (PORTFE)
$YFFA3A	NOT USED	PORT F EDGE-DETECT ENABLE (PFEER)
$YFFA3C	PORT F LEVEL (PFLVR)	PORT F INTERRUPT VECTOR (PFIVR)
$YFFA3E	RESERVED	
$YFFA40	TEST MODULE MASTER SHIFT A (TSTMSRA)	
$YFFA42	TEST MODULE MASTER SHIFT B (TSTMSRB)	
$YFFA44	TEST MODULE SHIFT COUNT (TSTSC)	
$YFFA46	TEST MODULE REPETITION COUNTER (TSTRC)	
$YFFA48	TEST MODULE CONTROL (CREG)	
$YFFA4A	TEST MODULE DISTRIBUTED REGISTER (DREG)	
$YFFA4C	RESERVED	

Table 7 SLIM Address Map (Continued)

Address	15 8	7 0
$YFFA4E	RESERVED	
$YFFA50	SYSTEM PROTECTION CONTROL (SYPCR)	
$YFFA52	TIMER CONTROL (TIC)	TIMER INTERRUPT VECTOR (TIV)
$YFFA54	NOT USED	SOFTWARE SERVICE (SWSR)
$YFFA56	PRESCALER (SLIMPRE – READ ONLY)	
$YFFA58	SOFTWARE WATCHDOG PERIOD REGISTER (SWP)	
$YFFA5A	REAL-TIME CLOCK PERIOD REGISTER (RTP)	
$YFFA5C	SOFTWARE WATCHDOG DOWNCOUNTER (SWDC – READ ONLY)	
$YFFA5E	REAL-TIME CLOCK DOWNCOUNTER (RTDC – READ ONLY)	
$YFFA60	CHIP-SELECT BASE ADDRESS REGISTER A (CSBARA)	
$YFFA62	CHIP-SELECT OPTION A (CSORA)	
$YFFA64	CHIP-SELECT BASE ADDRESS REGISTER B (CSBARB)	
$YFFA66	CHIP-SELECT OPTION B (CSORB)	
$YFFA68	CHIP-SELECT BASE ADDRESS REGISTER C (CSBARC)	
$YFFA6A	CHIP-SELECT OPTION C (CSORC)	
$YFFA6C	CHIP-SELECT CONTROL REGISTER (CSCR)	
$YFFA6E	RESERVED	
$YFFA70	RESERVED	
$YFFA72	RESERVED	
$YFFA74	RESERVED	
$YFFA76	RESERVED	
$YFFA78	RESERVED	
$YFFA7A	RESERVED	
$YFFA7C	RESERVED	
$YFFA7E	RESERVED	

1. Y = M111, where M is the logic state of the module mapping (MM) bit in the SLIMCR.

3.2 System Configuration

The SLIM controls MCU configuration during normal operation and during internal testing. The following section provides a description of the SLIM configuration registers and their contents.

System configuration registers are used for controlling initial bus configuration, clocking options, I/O port configuration, and selectively enabling/disabling SLIM submodules to reduce power consumption.

Additionally, the SLIM has a 512-byte (256 word) bootstrap ROM which can be used to configure the reset and interrupt vectors.

SLIMCR — SLIM Module Configuration Register **$YFFA00**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LOAD	FRZ[1:0]		RSVD[1]	MODE[2:0]			SHEN	SUPV	MM	ROMSU	ROMEN	IARB[3:0]			
RESET:															
$\overline{DRCD}$ = 0															
MCRC15	1	1	0	$\overline{DS}$	SIZE	R/$\overline{W}$	0	MCRC7	MCRC6	MCRC5	ADDR11	MCRC[3:0]			
$\overline{DRCD}$ = 1															
MCRC15	1	1	0	MCRC[11:9]			0	MCRC7	MCRC6	MCRC5	MCRC4	MCRC[3:0]			

1. This bit is reserved for future use.

The module configuration register (SLIMCR) controls system configuration. The reset state of some of the bits in SLIMCR depend on the state of corresponding bits in the module configuration shadow register (MCRC).

LOAD — Pad Driver Load

This bit controls the drive strength of all pad output drivers by selecting whether one or two pad drivers are enabled. This bit cannot be written and is controlled by the state of MCRC15.

0 = One pad driver is enabled.
1 = Two pad drivers are enabled.

FRZ[1:0] — FREEZE Assertion Response

Table 8 shows the effects of the assertion of FREEZE on the software watchdog, the real-time clock, and the prescaler. These bits can be written at any time.

Table 8 Effects of FREEZE Assertion

FRZ[1:0]		Disabled Elements
0	0	None
0	1	Bus monitor
1	0	Software watchdog, real-time clock, prescaler
1	1	Both

MODE[2:0] — Mode Configuration

These bits control the operating mode configuration of the external address and data bus, as shown in **Table 9**. These bits cannot be written and are controlled during reset by MCRC[11:9] if $\overline{DCRD}$ = 1, or by $\overline{DS}$, SIZE, and R/$\overline{W}$ if $\overline{DRCD}$ = 0.

Table 9 Mode Configuration Bit Field

MODE[2:0]			Configuration
0	X	X	Reserved
1	0	0	Expanded non-multiplexed mode
1	0	1	Expanded multiplexed mode
1	1	X	Single-chip mode

SHEN — Show Cycles Enable

This bit determines whether or not the EBI drives the data bus and the $\overline{AS}$, $\overline{DS}$, and R/$\overline{W}$ control signals during internal transfer operations.

NOTE

The address bus is always driven during internal cycles unless the ADRDIS bit is set to one in the chip-select control register (CSCR). This bit can be written at any time.

0 = Show cycles disabled
1 = Show cycles enabled. Data bus, $\overline{AS}$, $\overline{DS}$, and R/$\overline{W}$ control signals are driven during internal transfer operations.

SUPV — Supervisor/Unrestricted Data Space

The MC68HC16V1 operates only in supervisor mode. SUPV has no effect.

MM — Module Mapping

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

1 = Internal modules are addressed from $FFF000 – $FFFFFF.

IMB address lines ADDR[23:20] follow the logic state of ADDR19. MM corresponds to IMB ADDR23. If it is cleared, the SLIM maps IMB modules into address space $7FF000 – $7FFFFF, which is inaccessible to the CPU. Modules remain inaccessible until reset occurs. MM can only be written once after reset. Initialization software should leave MM set to logic level 1.

ROMSU — ROM Supervisor/Unrestricted Mode

The MC68HC16V1 operates only in supervisor mode. ROMSU has no effect.

ROMEN — ROM Enable

Allows boot ROM to respond to the 512-byte address range following $000000.

0 = Boot ROM does not respond and allows another internal module or external device to be mapped starting at address $000000. After reset, ROMEN = 0 also enables $\overline{CSB}$.

1 = Boot ROM responds to lower-order addresses.

IARB[3:0] — Interrupt Arbitration Field

Each module that can generate interrupt requests has an interrupt arbitration (IARB) field. Arbitration between interrupt requests of the same priority is performed by serial contention between IARB field bit values. Contention must take place whenever an interrupt request is acknowledged, even when there is only a single pending request.

An IARB field must have a non-zero value for contention to take place. If an interrupt request from a module with an IARB field value of %0000 is recognized, the CPU16 processes a spurious interrupt exception.

Because the SLIM routes external interrupt requests to the CPU16, the SLIM IARB field value is used for arbitration between internal and external interrupts of the same priority. The reset value of IARB for the SLIM is %1111 (highest priority), and the reset IARB value for all other modules is %0000, which prevents SLIM interrupts from being discarded during initialization. These bits can be written at any time.

MCRC — Module Configuration Shadow Register **$YFFA0A**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LOAD	0	0	RSVD[1]	MODE[2:0]			0	SUPV	MM	ROMSU	ROMEN	IARB[3:0]			
RESET:															
USER SPECIFIED															

1. This bit is reserved for future use.

The module configuration shadow register (MCRC) is a mask programmable register which contains the default reset values of corresponding bits in SLIMCR. During reset, SLIMCR is loaded from this register unless the drive reset configuration data ($\overline{DRCD}$) pin is asserted. If $\overline{DRCD}$ is asserted during reset, some MCRC shadow bits are overridden by the state of certain SLIM pins during reset. Refer to **Table 1** for the factory specified MCRC value appropriate to the MCU package being used.

SMD — Sub-Module Disable Register **$YFFA0D**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UNUSED												BRKEN	CHIP SELECT	TEST	SYS PROT
RESET:															
												0	0	0	0

The sub-module disable register (SMD) is used to disable sub-modules within the SLIM when they are not in use. A sub-module is disabled when its corresponding bit is set to one. Disabling the sub-module prevents it from being clocked, thereby placing the sub-module in its lowest power consumption state.

BRKEN — Breakpoint Enable

When enabled, this bit asserts $\overline{\text{BKPT}}$ on the IMB when any of the chip-select match signals is true, placing the CPU16 into background debug mode. This allows the user to breakpoint on addresses and conditions specified in the chip-select registers.

0 = Breakpoint logic is disabled
1 = Breakpoint logic is enabled

CHIP SELECT — Chip-Select Disable

0 = Chip-select sub-module is enabled
1 = Chip-select sub-module is disabled

TEST —Test Disable

0 = Test sub-module is enabled
1 = Test sub-module is disabled

SYSPROT— System Protect Disable

Disabling the system protection sub-module prevents the software watchdog timer, real time clock, prescaler, bus monitor, spurious interrupt monitor, and halt monitor from functioning.

0 = System protection sub-module is enabled
1 = System protection sub-module is disabled

PCON — Port/Clock Configuration Shadow Register **$YFFA0E**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	PCON[13:11]			PCON[10:9]		PCON[8:7]		PCON[6:5]		PCON4	PCON3	PCON2	PCON1	PCON0
RESET:		USER SPECIFIED													

The port/clock configuration shadow register (PCON) is a mask programmable register that determines the default reset value of corresponding bits and fields in the port C, D, and E pin assignment registers, as well as the default clock configuration. Refer to **Table 10**.

During reset, the specified configuration is loaded based on the information programmed in PCON at mask time. Driving $\overline{\text{DRCD}}$ to a logic level zero when $\overline{\text{RESET}}$ is asserted overrides the configuration in PCON for some port pin assignment bits and the clock select. This allows them to be configured from outside the device by the state driven on certain SLIM pins during reset. Not all bits in the port pin assignment registers have corresponding shadow bits in PCON.

NOTE

PCON programming must be consistent with pin availability. If a pin does not exist on the device, PCON bits should be programmed to configure the pin function as digital I/O out of reset, since EBI operation can be affected by the presence or absence of these pins. Refer to **Table 1** for the factory specified PCON value appropriate to the MCU package being used.

Table 10 Port/Clock Configuration Shadow Register (PCON)

Bit(s)[1]	States	Register	Pin(s) Affected	Selected Function
PCON[15:14]	0 0	NOT USED		
PCON[13:11]	0 0 0 0 0 1 0 1 0 0 1 1 1 X X	PCPAR[2:0]	ADDR[17:16]/PC[1:0]	PC[1:0] PC1, ADDR16 ADDR[17:16] Reserved Reserved
PCON10	1 0	PDPAR7	$\overline{CSA}$/PD7	$\overline{CSA}$ PD7
PCON9	1 0	CSCR5	PH[7:0]	$\overline{CSA}$ = 16-Bit Port $\overline{CSA}$ = 8-Bit Port
PCON8	1 0	PDPAR6	$\overline{CSB}$/PD6	$\overline{CSB}$ PD6
PCON7	1 0	CSCR4	PH[7:0]	$\overline{CSB}$ = 16-Bit Port $\overline{CSB}$ = 8-Bit Port
PCON6	1 0	PDPAR5	$\overline{CSC}$/PD5	$\overline{CSC}$ PD5
PCON5	1 0	CSCR3	PH[7:0]	$\overline{CSC}$ = 16-Bit Port $\overline{CSC}$ = 8-Bit Port
PCON4	1 0	PDPAR4	$\overline{DTACK}$/PD4	$\overline{DTACK}$ PD4
PCON3	1 0	PEPAR[7:5]	FC[2:0]/PE[7:5]	FC[2:0] PE[7:5]
PCON2	1 0	SYNCR[15, 13:12]	NONE	System Clock = 2 X Reference System Clock = 1 X Reference
PCON1	1 0	PEPAR4	CLKOUT/PE4	CLKOUT PE4
PCON0	1 0	—	NONE	Slow Reference Mode Fast Reference Mode

1. Certain PCON bit combinations are reserved in this SLIM implementation and should remain set to zero.

PCON[13:11] — Port C[1:0] Default Configuration

This bit field selects the number of pins in PC[1:0] which come out of reset configured as address lines. The address lines selected form a contiguous address space with ADDR[15:0]. The remaining pins in PC[1:0] come out of reset as digital output pins. The default reset value in PCON[13:11] is overridden if the SLIM comes out of reset in single chip mode. In this case, PC[1:0] are configured as digital inputs.

PCON[10:5] — Port D[7:5] Default Configuration

PCON10, PCON8, and PCON6 determine if $\overline{CSA}$, $\overline{CSB}$, and $\overline{CSC}$ are configured as chip-selects or digital I/O in the corresponding fields of PDPAR.

PCON9, PCON7, and PCON5 determine the port width if the associated chip-select is enabled.

PCON4 — Port D4 Default Configuration

This bit contains the default reset value of PDPAR4 to select the $\overline{DTACK}$/PD4 pin as $\overline{DTACK}$ or digital I/O out of reset.

PCON3 — Port E[7:5] Default Configuration

This bit controls whether PE[7:5] are configured as the three function code pins (FC[2:0]) or digital I/O pins.

0 = Pins configured as PE[7:5]
1 = Pins configured as FC[2:0]

PCON2 — 2X/1X System Clock

This bit controls the W and X bit reset states when the SLIM is reset in fast reference mode.

0 = System clock frequency is the same as the reference frequency.

1 = System clock frequency is twice the reference frequency.

This bit can be overridden by driving $\overline{IRQX}$ to the desired state if $\overline{DRCD}$ is low during reset.

PCON1 — Port E4 Default Configuration

This bit contains the default reset value for the CLKOUT/PE4 pin assignment bit in the port E pin assignment register (PEPAR). This allows the pin to come out of reset as CLKOUT or digital I/O.

PCON0 — Clock Select

This bit, along with $V_{DDSYN} = V_{DD}$, selects the clock source as either fast or slow reference.

0 = Clock source is fast reference

1 = Clock source is slow reference

If $V_{DDSYN} = V_{SS}$, the clock source is external, regardless of the state of this bit. This bit can be overridden by driving FREEZE to the desired state if $\overline{DRCD}$ is low during reset.

BOOT ROM — Reset and Interrupt Vector Configuration **$000000 – $0001FF**

The boot ROM submodule in the SLIM provides the system with a small amount of mask programmable memory used for initialization. The boot ROM submodule occupies 512 bytes (256 by 16 bits) of the system memory and is accessible by byte or word. The ROMEN and ROMSU bits in SLIMCR control access to the boot ROM. The contents of the boot ROM are user specified. The boot ROM on generic devices will be filled with all $FF or all $00. Consult the factory for information on ordering devices with a customer specified boot ROM.

3.3 System Clock

The system clock in the SLIM provides timing signals for the IMB modules and for the external bus interface. Because the MCU is a fully static design, register and memory contents are not affected when the clock rate changes. System hardware and software support changes in clock rate during operation.

3.3.1 System Clock Sources

The system clock signal can be generated from one of three sources. An internal phase-locked loop (PLL) can synthesize the clock from a slow or fast reference, or the clock signal can be input from an external frequency source. The slow reference typically is a 32.768 kHz crystal; the fast reference is typically a 4.194 MHz crystal. The slow and fast references may be generated by sources other than a crystal. Keep these clock sources in mind while reading the rest of this section.

The source of the system clock is determined at reset by the state of two shadow bits in the port/clock configuration register (PCON2 and PCON0) and the V_{DDSYN} pin. **Table 11** shows system clock sources.

The parameter "f_{ref}" refers to the frequency of the clock source connected to the EXTAL pin. Refer to **3.3.2 Clock Modes** for more information.

The parameter "f_{sys}" refers to the operating frequency of the MCU and has a defined relationship to f_{ref} that depends on the clock operating mode selected during reset.

Table 11 Clock Sources

MODE	SYNCR X/W/Y Bit Assignments	X/W/Y Reset Values	Clock Equation
External Clock V_{DDSYN} = 0 PCON0 = X[1]	X Y[2:0]	X = 1 Y = 0	$f_{sys} = \frac{f_{ref}}{(2-X)(2^{Y})}$, for $Y \leq 6$
Fast Reference V_{DDSYN} = 1 PCON0 = 0	X W[2:0] Y[2:0]	PCON2 = 0[2] X = 1 W = 0 Y = 0	$f_{sys} = \frac{f_{ref}(W+1)}{(2-X)(2^{Y})}$, for $Y \leq 6$
		PCON2 = 1[3] X = 0 W = 3 Y = 0	
Slow Reference V_{DDSYN} = 1 PCON0 = 1	X W Y[5:0]	X = 0 W = 0 Y = 63	$f_{sys} = 4f_{ref}(Y+1)(2^{(2W+X)})$

1. In external clock mode, the state of PCON0 has no effect on clock operation.
2. The default reset state mask-programmed in the PCON0 shadow bit can be overridden by the state of the FREEZE pin during reset when the $\overline{DRCD}$ pin is held low.
3. The default reset state mask-programmed in the PCON2 shadow bit can be overridden by the state of the $\overline{IRQX}$ pin during reset, if the $\overline{DRCD}$ pin is held low.

When V_{DDSYN} is held high during reset, the clock synthesizer generates a clock signal from either a crystal oscillator or an external reference input. The clock synthesizer control register (SYNCR) determines operating frequency and various clock options. When V_{DDSYN} is held low during reset, the clock synthesizer is disabled, and an external system clock signal must be applied.

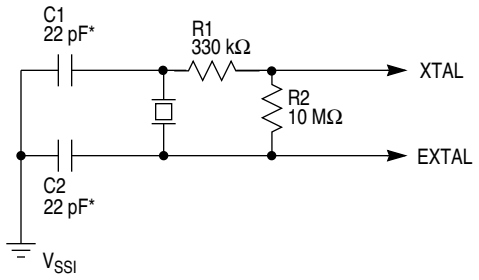
3.3.2 Clock Modes

In slow reference mode, the system clock is generated on-chip by a phase locked loop (PLL) frequency synthesizer and from an external reference (f_{ref}) that is typically 32.768 kHz. The frequency of the system clock is controlled by programming the X, Y, and W bits in the SYNCR.

NOTE

In slow reference mode, the f_{ref} input rate is not restricted to a single value (typically 32.768 kHz), but can range from 25 kHz to 50 kHz.

To generate the system clock using a slow reference, a crystal must be connected between the EXTAL and XTAL pins. **Figure 5** shows a typical circuit.



* RESISTANCE AND CAPACITANCE BASED ON A TEST CIRCUIT CONSTRUCTED WITH A KDS (DAISHINKU) DMX-38 32.768 kHz CRYSTAL. SPECIFIC COMPONENTS MUST BE BASED ON CRYSTAL TYPE. CONTACT CRYSTAL VENDOR FOR EXACT CIRCUIT.

16 OSCILLATOR

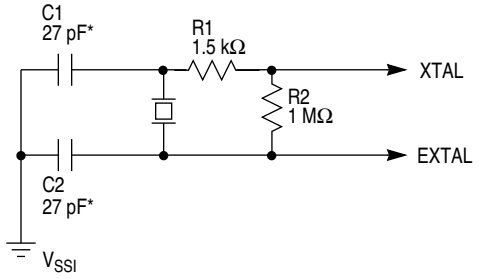
Figure 5 Slow Reference Crystal Circuit

In fast reference mode, the system clock is generated by the PLL from an external reference (f_{ref}) which is typically 4.194 MHz. The system clock frequency at reset is determined by the PCON2 shadow bit. Fast reference mode reduces jitter over the slow reference mode and, like slow reference mode, provides a 50% duty cycle system clock regardless of the reference duty cycle.

NOTE

In fast reference mode, the f_{ref} input rate is not restricted to a single value (typically 4.194 MHz), but can range from 1 MHz to 6 MHz.

To generate the system clock using a fast reference, a crystal must be connected between the EXTAL and XTAL pins. **Figure 6** shows a typical circuit.



* RESISTANCE AND CAPACITANCE BASED ON A TEST CIRCUIT CONSTRUCTED WITH A KDS041-18 4.194 MHz CRYSTAL. SPECIFIC COMPONENTS MUST BE BASED ON CRYSTAL TYPE. CONTACT CRYSTAL VENDOR FOR EXACT CIRCUIT.

16 OSCILLATOR 4M

Figure 6 Fast Reference Crystal Circuit

In external clock mode (when a crystal is not used), the clock source must be driven onto the EXTAL pin. This clock is used to generate the system clock directly (the PLL is turned off). At reset, the system clock frequency is the same as the input clock frequency (f_{ref}) and no dividers are present. If this frequency is the same as the maximum specified system clock frequency, it must not violate strict minimum duty cycle requirements. The duty cycle of the input is critical, especially at near maximum operating frequencies.

The relationship between clock signal duty cycle and clock signal period is expressed as follows:

$$\text{Minimum External Clock Period} = \frac{\text{Minimum External Clock High/Low Time}}{50\ \% - \text{Percentage Variation of External Clock Input Duty Cycle}}$$

When an externally generated reference signal (as opposed to a crystal) is applied through the EXTAL pin, regardless of the clock mode chosen, the XTAL pin must be left floating.

3.3.3 Clock Synthesizer Operation

V_{DDSYN} is used to power the clock circuits when the system clock is synthesized from a slow or fast reference. A separate power source increases MCU noise immunity and can be used to run the clock when the MCU is powered down. A quiet power supply must be used as the V_{DDSYN} source. Adequate external bypass capacitors should be placed as close as possible to the V_{DDSYN} pin to assure stable operating frequency. When an external system clock signal is applied and the PLL is disabled, V_{DDSYN} should be connected to the V_{SS} supply.

A voltage controlled oscillator (VCO) in the PLL generates the system clock signal. To maintain a 50% clock duty cycle, the internal VCO frequency is either two or four times the system clock frequency, depending on the state of the X bit in SYNCR. The clock signal is fed back to a divider/counter. The divider controls the frequency of one input to a phase comparator. The other phase comparator input is a reference signal, either from the crystal oscillator or from an external source. The comparator generates a control signal proportional to the difference in phase between the two inputs. This signal is routed through a low-pass filter and used to correct VCO output frequency.

Filter circuit implementation can vary, depending upon the external environment, reference multiplication factor, and required clock stability. **Figure 7** shows two recommended system clock filter networks for slow and fast reference modes. XFC pin leakage must be kept as low as possible to maintain optimum stability and PLL performance.

An external filter network connected to the XFC pin is not required when an external system clock signal is applied and the PLL is disabled (V_{DDSYN} = 0). The XFC pin must be left floating in this case.

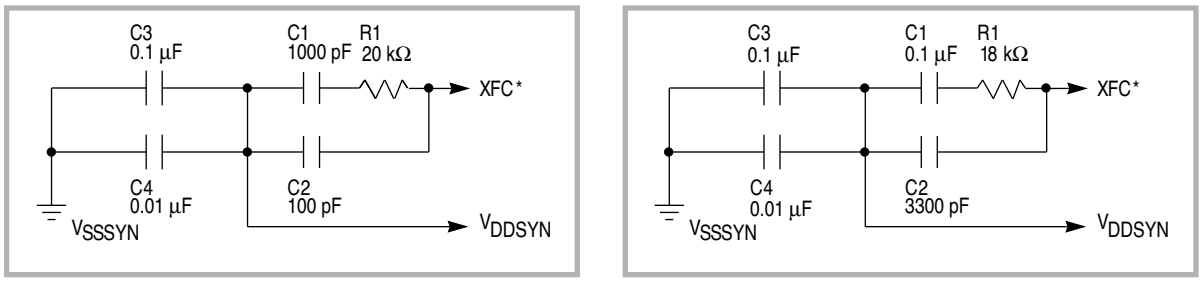


Figure 7 System Clock Filter Networks

The synthesizer locks when the divided VCO frequency is equal to f_{ref}. Lock time is affected by the filter time constant and by the amount of difference between the two comparator inputs. Whenever a comparator input changes, the synthesizer must relock. Lock status is shown by the SLOCK bit in SYNCR. During power-up, the MCU does not come out of reset until the synthesizer locks. Crystal type, characteristic frequency, and layout of external oscillator circuitry affect lock time.

The reset clock configuration involving V_{DDSYN} and PCON0 determines the clock mode selected. The functionality of SYNCR is affected by these configuration options.

In slow reference mode, one W bit and six Y bits are located in the PLL feedback path, enabling frequency multiplication by a factor of up to 256. The X bit is located in the VCO clock output path to enable dividing the system clock frequency by two without disturbing the PLL. Clock frequency is determined by the following equation:

$$f_{sys} = 4f_{ref}(Y + 1)(2^{(2W + X)})$$

In fast reference mode, three W bits are located in the PLL feedback path, enabling frequency multiplication by a factor from one to eight. Three Y bits and the X bit are located in the VCO clock output path to provide the ability to slow the system clock without disturbing the PLL. Clock frequency is determined by the following equation:

$$f_{sys} = \frac{f_{ref}(W + 1)}{(2 - X)(2^{Y})}, \text{ for } Y \leq 6$$

In external clock mode, three Y bits and the X bit are located between the EXTAL input and the system clock, to allow slowing the clock for reduced power consumption. Clock frequency is determined by the following equation:

$$f_{sys} = \frac{f_{ref}}{(2 - X)(2^{Y})}, \text{ for } Y \leq 6$$

For the device to perform correctly, the clock frequency selected by the W, X, and Y bits must be within the limits specified for the MCU.

Internal VCO frequency is determined by the following equations:

$$f_{VCO} = 4f_{sys} \text{ if } X = 0$$

or

$$f_{VCO} = 2f_{sys} \text{ if } X = 1$$

3.3.4 Clock Control

The clock control circuits determine system clock frequency and clock operation under special circumstances, such as following loss of synthesizer reference or during low-power operation. Clock source is determined by the logic state of the two shadow bits in the port/clock configuration register (PCON2 and PCON0) and V_{DDSYN} pin during reset.

SYNCR — Synthesizer Control Register **$YFFA04**

Slow Reference Mode

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
X	W	Y[5:0]						EDIV	STOSC	LOSCD	SLIMP	SLOCK	RSTEN	STSLIM	STEXT
RESET:															
0	0	1	1	1	1	1	1	0	0	0	0	1	0	0	0

Fast Reference Mode

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
X	W[2:0]			RSVD	Y[2:0]			EDIV	STOSC	LOSCD	SLIMP	SLOCK	RSTEN	STSLIM	STEXT
RESET:															
$\overline{\text{PCON2}}$	0	PCON2	PCON2	0	0	0	0	0	0	0	0	1	0	0	0

External Clock Mode

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
X	RSVD				Y[2:0]			EDIV	STOSC	LOSCD	SLIMP	SLOCK	RSTEN	STSLIM	STEXT
RESET:															
1	0	0	0	0	1	1	1	0	0	0	0	1	0	0	0

The synthesizer control register (SYNCR) can be read or written at any time. The encoding and default value of the upper byte of SYNCR depend on the clock mode selected at reset.

For slow reference mode, the default value is $3F, which corresponds to an operating frequency of 256 times f_{ref}. For a 32.768 kHz crystal, the operating frequency is 8.388 MHz.

In fast reference mode, with PCON2 cleared to zero, the default value of the upper byte is $80, which corresponds to a 1-to-1 match of the reference frequency. With PCON2 set to one, the default value of the upper byte is $30, which corresponds to an operating frequency of two times f_{ref}. For a 4.194 MHz crystal, the operating frequency is 8.388 MHz.

In external clock mode, the default value of the upper byte is $80, which corresponds to an operating frequency which is equal to the input frequency on EXTAL. Strict minimum duty cycle requirements apply. Refer to **3.3.2 Clock Modes** for more information. The default value is forced into the SYNCR upon reset, along with the default values of the other bits in the register.

X — Frequency Control Bit

This bit controls a one-bit divider which drives the system clock in all modes. When X is set, the divider is bypassed; when clear, the system clock is divided by two.

W — Frequency Control Bit(s)

This field has different lengths and functions, depending on the clock mode. In slow reference mode, a single W bit is used to multiply the reference frequency. In fast reference mode, three W bits are used to multiply the reference frequency.

Y — Frequency Control Bit(s)

This field has different lengths and functions, depending on the clock mode. In slow reference mode, six Y bits are used to multiply the reference frequency. In fast reference mode, three Y bits are used to divide the PLL output frequency. In the external clock mode, three Y bits are used to divide the input clock frequency.

EDIV — E Clock Divide Rate

0 = ECLK frequency is system clock divided by 8.
1 = ECLK frequency is system clock divided by 16.

ECLK is an external M6800 bus clock available on pin ADDR23 of some SLIM implementations. EDIV also affects ECLK synchronized chip-select bus cycles. Refer to **3.9 Chip-Selects** for more information. The MC68HC16V1 does not have the ADDR23/ECLK pin, however, ECLK synchronized chip-selects may still be used.

STOSC — Stop Oscillator

0 = The crystal oscillator circuit continues to operate while in LPSTOP.
1 = The crystal oscillator circuit and the PLL are turned off to save power during LPSTOP. All clocks on the chip are stopped. In LPSTOP mode, all SLIM interrupt request lines become asynchronous, and all system protection functions are disabled.

The STOSC bit has no effect on the system clock in external clock mode.

LOSCD — Loss of Clock Oscillator Disable

0 = Loss of clock oscillator reference is enabled.
1 = Loss of clock oscillator reference is disabled and not running.

SLIMP — Limp Mode

0 = System clock is being provided normally, either by the PLL or by an external clock from the EXTAL input.
1 = Loss of system clock has been detected, and the system clock is being provided from the loss of clock oscillator reference.

SLOCK — Synthesizer Lock Flag

0 = VCO is enabled, but has not yet locked into the narrow bandwidth mode.
1 = VCO is disabled (system clock is driven in directly), or the VCO is locked.

RSTEN — Reset Enable

0 = Loss of clock causes the MCU to operate in limp mode.
1 = Loss of clock causes reset.

STSLIM — Stop Mode SLIM Clock

0 = When LPSTOP is executed, the SLIM clock is driven from the crystal oscillator and the VCO is turned off to conserve power.
1 = When LPSTOP is executed, the SLIM clock is driven from the VCO.

STEXT — Stop Mode External Clock

0 = When LPSTOP is executed, no external clock is driven.
1 = When LPSTOP is executed, the external clock is driven from the SLIM clock, as determined by the state of the STSLIM bit.

3.3.5 Avoiding Frequency Overshoot

In slow and fast reference modes, changing the W and Y fields in the SYNCR to increase the operating frequency can result in frequency overshoot. This overshoot can be avoided by using the following procedure:

1. Determine the values for the W and Y fields which will result in the desired frequency when the X bit is set.
2. With the X bit cleared, write these values for the W and Y fields to SYNCR.
3. After the VCO locks, set the X bit in the SYNCR. This changes the clock frequency to the desired frequency.

For example, use the following procedure in fast reference mode (PCON2 = 1) to change the clock frequency from 8.388 MHz to 20.972 MHz after reset using a 4.194 MHz crystal:

1. Determine the values for the W and Y fields: W = %100, Y = %000
2. Write the values to SYNCR: W = %100, Y = %000
3. Wait for the SYNCR SLOCK bit to set. When SLOCK = 1, the VCO has locked the new frequency, now 10.486 MHz.
4. Set the SYNCR X bit to one. This removes a divider from the synthesizer output path, effectively doubling the clock frequency from 10.486 MHz to 20.972 MHz.

3.4 System Protection

The system protection block incorporates several monitoring and timing functions. System protection block features include:

- Software watchdog timer
- Real-time clock
- Prescaler
- Bus monitor
- Spurious interrupt monitor
- Halt monitor

These functions have been made integral to the microcontroller to reduce the number of external components in a complete control system.

Figure 8 shows the system protection block.

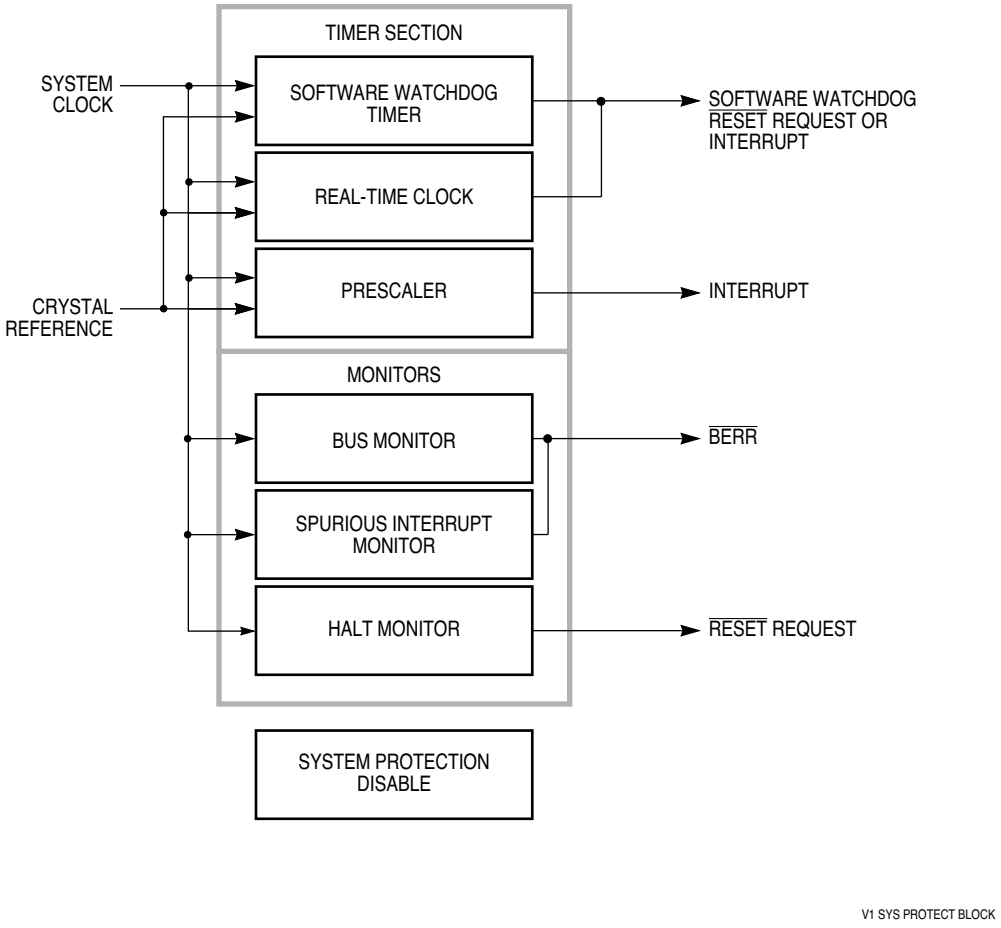


Figure 8 System Protection Block

Figure 9 shows the configuration of the software watchdog, the real-time clock and the prescaler.

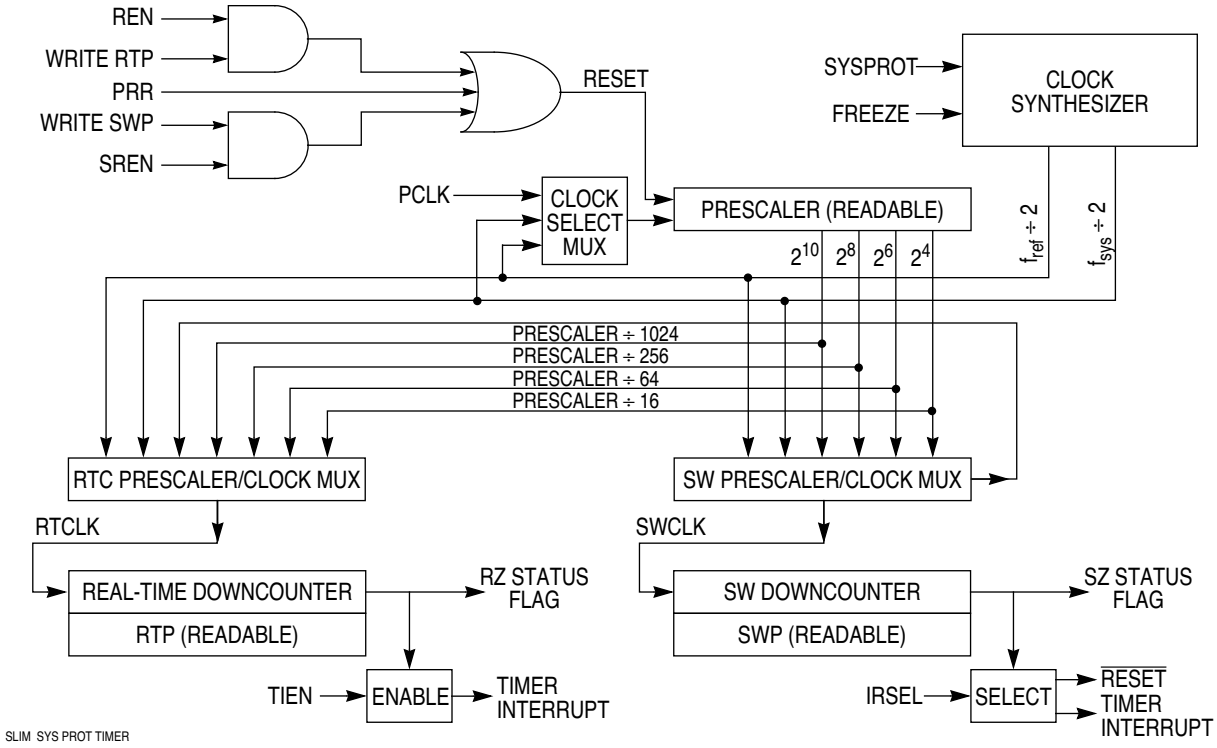


Figure 9 Software Watchdog, Real-Time Clock and Prescaler

SYPCR — System Protection Control Register **$YFFA50**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REN	SREN	PCLK	SLPC	RZ	SZ	IRSEL	TIEN	0	TIQL[2:0]			HME	BME	BMT[1:0]	
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The system protection control register controls system monitor functions such as the halt monitor, the software watchdog, and the bus monitor.

REN — Real-Time Reset Enable

This bit controls whether writes to the RTP register reset the prescaler.

0 = Prescaler not reset when writing to RTP.

1 = Prescaler reset when writing to RTP.

SREN — Software Watchdog Reset Enable

This bit controls whether writes to the SWP register reset the prescaler.

0 = Prescaler not reset when writing to the SWP.

1 = Prescaler reset when writing to the SWP.

PCLK — Prescaler Clock Select

This bit selects the clock source to the prescaler.

0 = Selects system clock output divided by two ($f_{sys} \div 2$).

1 = Selects clock synthesizer reference input divided by two ($f_{ref} \div 2$).

SLPC — Software Watchdog LPSTOP Count Select

0 = Software watchdog does not count in LPSTOP; counting is resumed after reset.
1 = Software watchdog counts in LPSTOP.

RZ — Timer Zero Flag

This bit is set after the RTC downcounter reaches zero. Clear the bit by reading the register with the RZ bit set to one, then writing the RZ bit to zero.

SZ— Software Watchdog Zero Flag

This bit is set after the software watchdog downcounter reaches zero. Clear the bit by reading the register with the SZ bit set to one, then writing the SZ bit to zero.

IRSEL — Interrupt/Reset Select

This bit selects whether the software watchdog downcounter issues a reset or an interrupt upon counting down to zero.

0 = Counter issues reset
1 = Counter issues the interrupt designated by TIQL[2:0]

TIEN — Timer Interrupt Enable

This bit controls whether the real-time down counter issues an interrupt after a timeout.

0 = No interrupt is issued upon reaching zero
1 = Interrupt is issued upon reaching zero

TIQL[2:0] — Timer Interrupt Request Level

This field defines the priority of the interrupt that is generated when the real-time down counter or the software watchdog down counter time out. Refer to **Table 12**.

Table 12 Timer Interrupt Request Level Field

TIQL[2:0]			Description
0	0	0	Interrupt Disabled
0	0	1	Interrupt Request Level 1
0	1	0	Interrupt Request Level 2
0	1	1	Interrupt Request Level 3
1	0	0	Interrupt Request Level 4
1	0	1	Interrupt Request Level 5
1	1	0	Interrupt Request Level 6
1	1	1	Interrupt Request Level 7

HME — Halt Monitor Enable

The HME bit controls the halt monitor as it checks for double bus faults on the IMB.

0 = Disable halt monitor function
1 = Enable halt monitor function

BME — Bus Monitor Enable

The BME bit enables the internal bus monitor for internal to external bus cycles. It can be written only once after reset.

0 = Disable bus monitor function for internal to external bus cycles.
1 = Enable bus monitor function for internal to external bus cycles.

BMT[1:0] — Bus Monitor Timing

This bit field selects the time-out period in system clocks for the bus monitor. It can be written only once after reset. Refer to **Table 13**.

Table 13 Bus Monitor Timeout Period

BMT[1:0]	Timeout
0 0	64 System Clocks
0 1	32 System Clocks
1 0	16 System Clocks
1 1	8 System Clocks

After reset, bus monitor timeout value defaults to 64 system clocks.

NOTE

For external accesses, this time-out value should be longer than the chip-select access time.

3.4.1 Bus Monitor

The internal bus monitor checks for excessively long response times during normal bus cycles ($\overline{\text{DTACK}}$). The monitor asserts the internal $\overline{\text{BERR}}$ signal if response time is excessive.

$\overline{\text{DTACK}}$ response time is measured in clock cycles. The maximum allowable response time can be selected by setting BMT[1:0].

The BME bit in SYPCR enables the internal bus monitor for internal to external bus cycles. The monitor does not check $\overline{\text{DTACK}}$ response on the external bus unless the CPU16 initiates the bus cycle. If a system contains external bus masters, an external bus monitor must be implemented and the internal to external bus monitor option must be disabled.

3.4.2 Spurious Interrupt Monitor

The spurious interrupt monitor issues internal $\overline{\text{BERR}}$ if no interrupt arbitration occurs during an IACK cycle. Leaving IARB[3:0] set to %0000 in the module configuration register of any peripheral that can generate interrupts will cause a spurious interrupt. Refer to **Table 49** for the CPU16 interrupt vector table.

3.4.3 Halt Monitor

The halt monitor responds to an assertion of the HALT signal on the internal bus caused by a double bus fault. This signal is asserted by the CPU16 after a double bus fault occurs. If the HME bit is set to one, the halt monitor is enabled, and asserts $\overline{\text{RESET}}$ when the HALT signal is asserted. A flag in the reset status register (RSR) indicates that the last reset was caused by the halt monitor. The halt monitor reset can be inhibited by clearing the HME bit in SYPCR.

3.4.4 Timer Control

The timer control register selects the time base for the real-time clock and software watchdog. It also sets the vector used when these timers generate interrupts.

TIC — Timer Control Register **$YFFA52**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PRR	SWC[2:0]			0	RTC[2:0]			TIV[7:0]							
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The timer control register selects the clock input to the software watchdog downcounter and the real-time clock downcounter.

PRR — Prescaler Reset

Writing TIC with PRR set to one resets the prescaler (PRE) when changing the time base for the software watchdog and/or the real-time clock.

0 = Prevents prescaler reset
1 = Permits prescaler reset

SWC[2:0] — Software Watchdog Clock Control

The SWC[2:0] field provides seven possible settings for choosing the input clock for the software watchdog downcounter. Refer to **Table 14**.

RTC[2:0] — Real-Time Clock Control

The RTC[2:0] field provides six possible settings for choosing the input clock for the real-time clock downcounter. Refer to **Table 14**.

Table 14 Software Watchdog and Real-Time Clock Control Settings

Clock Control Setting			SWC[2:0]	RTC[2:0]
0	0	0	$f_{sys}/2$	$f_{sys}/2$
0	0	1	$f_{ref}/2$	$f_{ref}/2$
0	1	0	$f_{pre}/1024$	$f_{pre}/1024$
0	1	1	$f_{pre}/256$	$f_{pre}/256$
1	0	0	$f_{pre}/64$	$f_{pre}/64$
1	0	1	$f_{pre}/16$	$f_{pre}/16$
1	1	0	RTC Extension	RTC Off
1	1	1	SWDOG Off	RTC Off

In the case of the SWC[2:0] field, the RTC extension setting allows the real-time downcounter and the software watchdog downcounter to operate in chained mode (refer to paragraph **3.4.7 Real-Time Clock and Software Watchdog in Chained Mode** for more information).

The system clock and the crystal reference settings allow clocking the SWDOG and/or the RTC downcounter by the crystal frequency divided by two, or the system clock divided by two, respectively. The prescaler settings, on the other hand, allow one of four prescaler taps to clock the downcounters.

The prescaler frequency (f_{pre}) is defined as follows based on the setting of the PCLK bit in SYPCR:

$$f_{pre} = f_{sys}/2 \text{ if PCLK} = 0$$

or

$$f_{pre} = f_{ref}/2 \text{ if PCLK} = 1$$

Resetting out of LPSTOP has no effect on either the SWC or RTC fields. In this way, the SWDOG and the RTC continue to run with the same clock configuration both in and out of LPSTOP.

TIV[7:0] — Timer Interrupt Vector

This bit field contains the vector number that is generated during an IACK cycle in response to a timer interrupt.

3.4.5 Software Watchdog

The software watchdog (SWDOG) monitors system software and protects against possible errors occurring in the system which might allow software to become trapped in loops with no controlled exit. User code must regularly write the servicing sequence to the SWSR to prevent the software watchdog from timing out. If this real-time servicing action does not take place, the software watchdog issues either a reset or an interrupt.

Software watchdog features include:

- The ability to select between an interrupt or a reset at the end of a time-out period.
- The input clock to the software downcounter (SWDC) may be the system clock divided by two ($f_{sys} \div 2$), the clock synthesizer reference frequency divided by 2 ($f_{ref} \div 2$), or the output of the system protection prescaler. Time-out periods range from 61 µs to 4096 seconds (with a 32.768 kHz crystal frequency) or from 95.4 ns to 6.4 seconds (with a 20.97 MHz external clock).
- SWDC can be chained together with the real-time downcounter to create a 32-bit downcounter for extended time-out periods.
- SWDC is readable at any time.

SWSR — Software Watchdog Service Register **$YFFA55**

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

The software watchdog service register (SWSR) is controlled by SWE in SYPCR. Once enabled, the watchdog requires that a service sequence be written to SWSR on a periodic basis. If servicing does not take place, the watchdog times out and issues a reset or generates an interrupt. This register can be written at any time, but returns zeros when read.

Each time the service sequence is written, the software watchdog timer restarts. The sequence to restart consists of the following steps:

- Write $55 to SWSR
- Write $AA to SWSR

Both writes must occur before time-out in the order listed, but any number of instructions, up to the end of the time-out period, can be executed between the two writes.

When the proper service sequence occurs, the software watchdog downcounter reloads the value of the software watchdog period register and the process begins again.

If the time-out period is reached before a service action, the software watchdog causes a reset, or an interrupt, depending on the state of the IRSEL bit. If the software watchdog causes reset, the SW flag in the reset status register (RSR) is also set.

SLIMPRE — System Protection Prescaler **$YFFA56**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UNUSED															
POWER-ON RESET ONLY:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The software watchdog and the real-time clock share a 10-bit synchronous counter, the prescaler (SLIMPRE). Different prescaler taps can be applied simultaneously to either the software watchdog or real-time clock downcounter. A read of unused bits always returns zero. Writes have no effect.

SWP — Software Watchdog Period Register **$YFFA58**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
POWER-ON RESET ONLY:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The software watchdog period register (SWP) contains the value that is loaded into the software watchdog downcounter. The lower byte of SWP must be written in order to trigger the software watchdog downcounter.

SWP is readable and writable at any time. Writes to this register reset the prescaler if the SREN bit in SYPCR is set.

A zero loaded into this register turns off and clears the software watchdog downcounter.

SWP is not affected by reset. Therefore, the value programmed into SWP before reset occurs is retained. Power-on resets SWP to zero.

The SWP count value is determined by the following equation:

$$\text{SWP Count Value} = \text{Desired SWDOG Time-out Period} \times \text{SWDOG Clock Source}$$

Table 15 shows the software watchdog period ranges.

Table 15 Software Watchdog Period Ranges (Timers Not Chained)[1]

SWC[2:0] Clock Control Settings			Clock Source	Time-Out Ranges for Typical Values of f_{ref} and f_{sys}[2]			
				32.768 kHz	4.194 MHz	16.777 MHz	20.972 MHz
0	0	0	$f_{sys}/2$	61.0 µs to 4.0 s	477 ns to 31.3 ms	119 ns to 7.81 ms	95.4 ns to 6.25 ms
0	0	1	$f_{ref}/2$	61.0 µs to 4.0 s	477 ns to 31.3 ms	119 ns to 7.81 ms	95.4 ns to 6.25 ms
0	1	0	$f_{pre}/1024$	62.5 ms to 68.3 mins	488 µs to 32 s	122 µs to 8 s	97.7 µs to 6.4 s
0	1	1	$f_{pre}/256$	15.6 ms to 17.1 mins	122 µs to 8 s	30.5 µs to 2 s	24.4 µs to 1.6 s
1	0	0	$f_{pre}/64$	3.91 ms to 4.27 mins	30.5 µs to 2 s	7.63 µs to 500 ms	6.10 µs to 400 ms
1	0	1	$f_{pre}/16$	977 µs to 64.0 s	7.63 µs to 500 ms	1.91 µs to 125 ms	1.53 µs to 100 ms
1	1	0	Chained Timer	—	—	—	—
1	1	1	Software Watchdog Off	—	—	—	—

1. Refer to **Table 17** for chained timer time-out period information.
2. These frequency categories are frequencies which are output from the clock select mux to the prescaler. The prescaler frequency is defined as $f_{pre} = f_{sys}/2$ if PCLK = 0 or $f_{pre} = f_{ref}/2$ if PCLK = 1. Refer to **Figure 9**.

SWDC — Software Watchdog Downcounter **$YFFA5C**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
POWER-ON RESET ONLY:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The software watchdog downcounter (SWDC) consists of a 16-bit downcounter clocked by the output of the prescaler multiplexer. This signal is applied to SWDC which counts down to zero when loaded with the SWP value. SWDC does not begin counting until the lower byte of SWP is written.

When the software watchdog times out, depending on the state of the IRSEL bit in SYPCR, it asserts a reset or generates an interrupt. SWDC is then loaded with the current time-out value from the SWP and begins counting down again. The time-out value may be changed at any time.

After timeout occurs, the software watchdog zero flag (SZ) in SYPCR is set.

SWDC is readable at any time.

When $\overline{\text{RESET}}$ is asserted, SWDC is not affected.

3.4.6 Real-Time Clock

The real-time clock (RTC) consists of a 16-bit downcounter clocked by the RTCLK output of the RTC prescaler multiplexer.

RTCLK is applied to a 16-bit downcounter that counts down to zero when loaded with the RTP value. RTDC does not start counting until the lower byte of RTP is written.

RTP — Real-Time Period Register **$YFFA5A**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
POWER-ON RESET ONLY:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The real-time period register (RTP) contains the value to be loaded into the RTC downcounter. The lower byte of RTP must be written in order to trigger the real-time downcounter, both when chained and when not chained.

Writes to this register reset the prescaler if the REN bit in SYPCR is set. A zero loaded into this register turns off and clears the RTC downcounter.

The RTP count value is determined by the following equation:

$$\text{RTP Count Value} = \text{Desired RTC Time-out Period} \times \text{RTC Clock Source}$$

Table 16 shows the real-time period ranges.

Table 16 Real-Time Period Ranges (Timers Not Chained)[1]

RTC[2:0] Clock Control Settings			Clock Source	Time-Out Ranges for Typical Values of f_{ref} and f_{sys}[2]			
				32.768 kHz	4.194 MHz	16.777 MHz	20.972 MHz
0	0	0	$f_{sys}/2$	61.0 µs to 4.0 s	477 ns to 31.3 ms	119 ns to 7.81 ms	95.4 ns to 6.25 ms
0	0	1	$f_{ref}/2$	61.0 µs to 4.0 s	477 ns to 31.3 ms	119 ns to 7.81 ms	95.4 ns to 6.25 ms
0	1	0	$f_{pre}/1024$	62.5 ms to 68.3 mins	488 µs to 32 s	122 µs to 8 s	97.7 µs to 6.4 s
0	1	1	$f_{pre}/256$	15.6 ms to 17.1 mins	122 µs to 8 s	30.5 µs to 2 s	24.4 µs to 1.6 s
1	0	0	$f_{pre}/64$	3.91 ms to 4.27 mins	30.5 µs to 2 s	7.63 µs to 500 ms	6.10 µs to 400 ms
1	0	1	$f_{pre}/16$	977 µs to 64.0 s	7.63 µs to 500 ms	1.91 µs to 125 ms	1.53 µs to 100 ms
1	1	0	Real-Time Clock Off	—	—	—	—
1	1	1	Real-Time Clock Off	—	—	—	—

1. Refer to **Table 17** for chained timer time-out period information.
2. These frequency categories are frequencies which are output from the clock select mux to the prescaler. The prescaler frequency is defined as $f_{pre} = f_{sys}/2$ if PCLK = 0 or $f_{pre} = f_{ref}/2$ if PCLK = 1. Refer to **Figure 9**.

RTDC — Real-Time Clock Downcounter **$YFFA5E**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
POWER-ON RESET ONLY:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

When the real-time clock downcounter (RTDC) reaches zero, the RZ flag in the SYPCR is set to one. An interrupt is generated if the TIEN bit is set and if TIQL[2:0] is non-zero. RTDC then loads the current RTP value and begins counting again. When $\overline{\text{RESET}}$ is asserted, RTDC is not affected.

3.4.7 Real-Time Clock and Software Watchdog in Chained Mode

The real-time clock and the software watchdog can be chained together to form a 32-bit downcounter. At this time, SWDC acts as the most significant word (MSW) and RTDC acts as the least significant word (LSW). Upon reaching zero, the least significant word, rather than reloading the period register, counts the SWDC down by one as it rolls over to $FFFF, and then continues counting.

One 32-bit or two 16-bit values can be written to the SWP and RTP registers to trigger the downcounter chain.

NOTE

The downcounter chain does not begin counting until the lower byte of RTP is written.

The counter can be read at any time. Anything other than 32-bit reads, however, may not be accurate.

The concatenated SWP/RTP count value is determined by the following equation:

$$\text{Concatenated SWP/RTP Count Value} = \text{Desired Period of Chained Timer} \times \text{RTC Clock Source}$$

Convert the result of the formula above into hexadecimal and place the MSW in the SWP register and the LSW in the RTP register.

These calculations give the real-time period ranges shown in **Table 17**.

Table 17 Timer Period Ranges (Downcounters Chained)

RTDC Clock Source	Time-Out Ranges for Typical Values of f_{ref} and f_{sys} [1]			
	32.768 kHz	4.194 MHz	16.777 MHz	20.972 MHz
$f_{sys}/2$ or $f_{ref}/2$	61.0 µs to 72.8 hrs	477 ns to 34.1 mins	119 ns to 8.53 mins	95.4 ns to 6.83 mins
$f_{pre}/16$	977 µs to 48.5 days	7.63 µs to 9.10 hrs	1.91 µs to 8102 s	1.53 µs to 1.82 hrs
$f_{pre}/64$	3.91 ms to 194 days	30.5 µs to 36.4 hrs	7.63 µs to 9.10 hrs	6.10 µs to 7.28 hrs
$f_{pre}/256$	15.6 ms to 2.13 yrs	122 µs to 6.07 days	30.5 µs to 36.4 hrs	24.4 µs to 29.1 hrs
$f_{pre}/1024$	62.5 ms to 8.51 yrs	488 µs to 24.3 days	122 µs to 6.07 days	97.7 µs to 4.85 days

1. The resolution of each range within a system frequency category is the range's lower bound. For example, in the first row of the 32.768 kHz column, the resolution is 61.0 µs.

3.5 External Bus Interface

The external bus interface (EBI) transfers information between the internal MCU bus and external devices. The external bus has 18 address lines and 16 data lines.

The EBI operates as a bus master and allows the CPU to access external memory resources. The EBI runs external bus cycles based on the programming of the chip-select base address and option registers, and the chip select pin options programmed in the port D pin assignment register.

The EBI supports byte, word, and long-word transfers. Ports are accessed through the use of asynchronous cycles controlled by the data size (SIZE) and data transfer acknowledge ($\overline{\text{DTACK}}$) pins.

External bus operation is synchronous to the clock output (CLKOUT) signal. The external bus speed is programmable from two to 16 clock cycles (0 to 14 wait states) with optional external bus termination using the $\overline{\text{DTACK}}$ signal. The bus speed is programmed on a per chip-select basis in the $\overline{\text{DTACK}}$ field of each chip-select option register (CSOR).

At least one chip-select must be programmed to provide external bus cycle termination unless bus cycles are to be terminated by the use of the $\overline{\text{DTACK}}$ function. Chip-selects used to access external memories or peripherals can be programmed for 8- or 16-bit, multiplexed or non-multiplexed accesses on a per chip-select basis. Operation of the boot chip-select, $\overline{\text{CSB}}$, can be overridden by the state of the ADDR/DATA[8:7] pins during reset. Refer to **3.6.2 Default Configuration Override During Reset** for more information.

3.5.1 Transfer Size

At the beginning of a bus cycle, the SIZE signal is driven along with the function code signals. The SIZE signal is only relevant for 16-bit port accesses and indicates the number of bytes remaining to be transferred during an operand cycle (refer to **Table 18**). It is valid while the address strobe ($\overline{\text{AS}}$) is asserted.

Table 18 SIZE Signal Encoding

SIZE	Transfer Size
1	Byte
0	Word

3.5.2 Function Codes

Function code signals (FC[2:0]) are automatically generated by the CPU16. The function codes can be considered address extensions that automatically select one of eight address spaces to which an address applies. These spaces are designated as either user or supervisor, and program or data spaces. Because the CPU16 always operates in supervisor mode (FC2 always = 1), address spaces 0 through 3 are not used. Address space 7 is designated CPU space. CPU space is used for control information not normally associated with read or write bus cycles. Function codes are valid while $\overline{AS}$ is asserted.

Table 19 is a summary of the function code assignments.

Table 19 CPU16 Address Space Encoding

FC2	FC1	FC0	Address Space
1	0	0	Reserved
1	0	1	Data Space
1	1	0	Program Space
1	1	1	CPU Space

3.5.3 Address Bus

Address bus signals (ADDR[17:0]) define the address of the most significant byte to be transferred during a bus cycle. The MCU places the address on the bus at the beginning of a bus cycle. The address is valid while $\overline{AS}$ is asserted.

3.5.4 Address Strobe

Address strobe ($\overline{AS}$) is a timing signal that indicates the validity of an address on the address bus and the validity of many control signals. It is asserted one-half clock after the beginning of a bus cycle. In multiplexed mode, a valid address is qualified by the ALE signal.

3.5.5 Data Bus

In non-multiplexed modes, data bus signals DATA[15:0] comprise a bidirectional parallel bus that transfers data to or from the MCU. A read or write operation can transfer 8 or 16 bits of data in one bus cycle. During a read cycle, the data is latched by the MCU on the last falling edge of the clock for that bus cycle. For a write cycle, all 16 bits of the data bus are driven, regardless of the port width or operand size. The MCU places the data on the data bus one-half clock cycle after $\overline{AS}$ is asserted in a write cycle.

In multiplexed mode, the 16-bit address/data bus contains valid addresses whenever address latch enable (ALE) is asserted. After ALE is negated, $\overline{DS}$ is asserted and data is transferred to or from the MCU.

3.5.6 Data Strobe

Data strobe ($\overline{DS}$) is a timing signal. For a read cycle, the MCU asserts $\overline{DS}$ to signal an external device to place data on the bus. $\overline{DS}$ is asserted at the same time as $\overline{AS}$ during a read cycle. For a write cycle, $\overline{DS}$ signals an external device that data on the bus is valid. The MCU asserts $\overline{DS}$ one full clock cycle after the assertion of $\overline{AS}$ during a write cycle.

3.5.7 Address Latch Enable

In multiplexed mode, address latch enable (ALE) denotes a valid address on the ADDR/DATA[15:0] bus. ALE asserts during the high time of the first CLKOUT cycle of each multiplexed bus access, and negates during the CLKOUT low time that immediately follows. **Figure 10** shows an example of how to demultiplex the address/data bus using ALE.

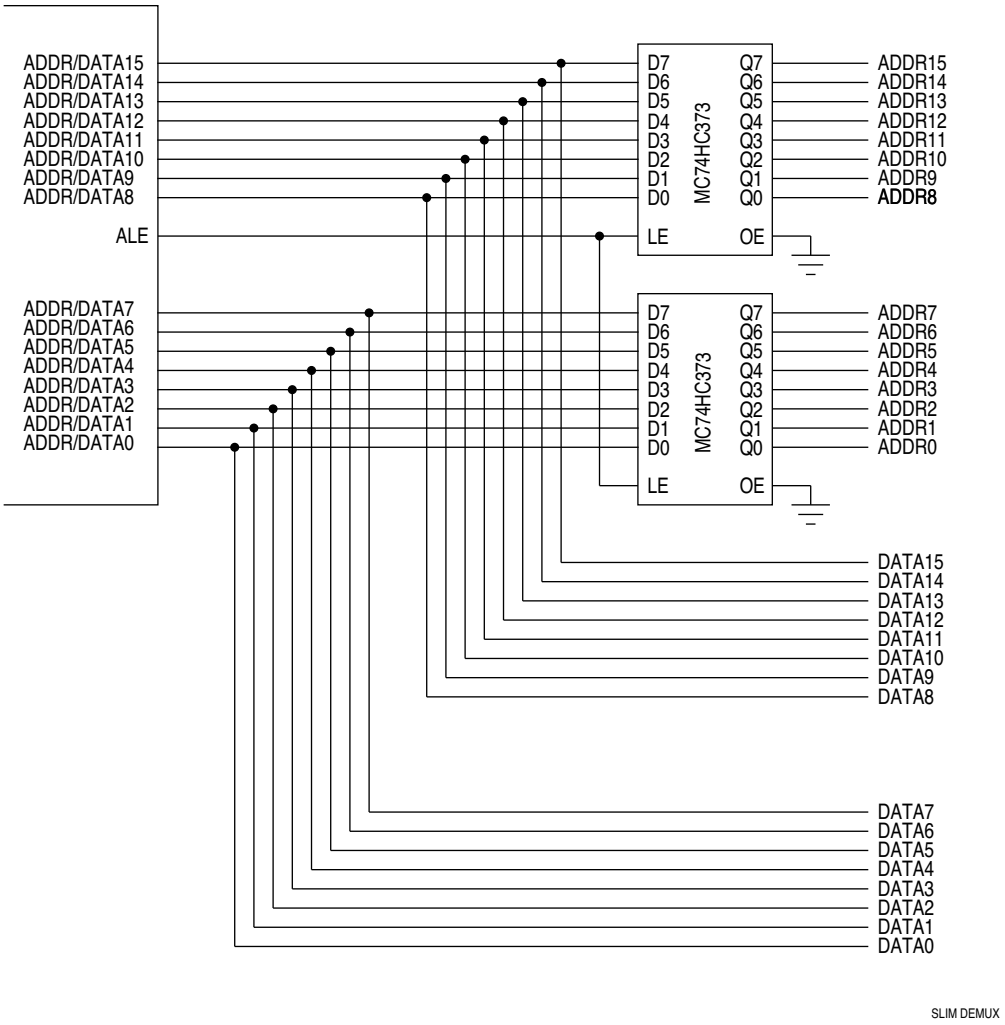


Figure 10 Address/Data Bus Demultiplexing

3.5.8 Read/Write

The read/write (R/$\overline{W}$) signal determines the direction of the transfer during a bus cycle. This signal changes state, when required, at the beginning of a bus cycle, and is valid while $\overline{AS}$ is asserted. R/$\overline{W}$ only transitions when a write cycle is preceded by a read cycle or vice versa. The signal can remain low for two consecutive write cycles.

3.5.9 Alternate Read and Write Strobes

When the RWEN bit in the chip-select control register (CSCR) is set to one, the $\overline{DS}$ and R/$\overline{W}$ pins become $\overline{RD}$ and $\overline{WR}$ respectively. $\overline{RD}$ and $\overline{WR}$ are alternate read and write strobe signals that can be used to simplify the interface to some memory and peripheral devices. **Figure 11** shows a typical circuit used to generate $\overline{RD}$ and $\overline{WR}$.

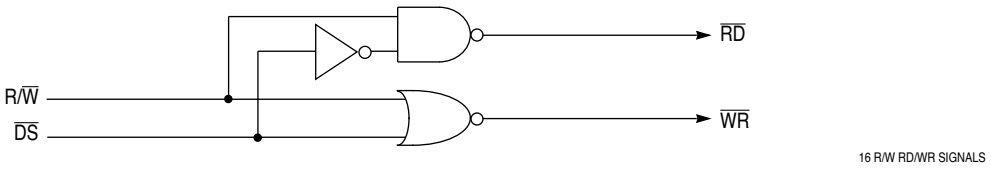


Figure 11 RD and WR Generation

3.5.10 Data Transfer Acknowledge

External bus cycles must be terminated by DTACK. DTACK can be supplied either by SLIM chip-select logic or by asserting the DTACK pin if it is configured for its primary function. Modules internal to the MCU supply their own DTACK after completing a bus cycle. During a read cycle, DTACK signals the MCU to terminate the bus cycle and latch data. During a write cycle, DTACK indicates that data has been successfully stored and that the cycle can end. If the DTACK pin is asserted before SLIM chip-select logic generates DTACK, the EBI terminates the bus cycle without waiting for the chip-select logic to do so.

3.5.11 Data Transfer Mechanism

The MCU architecture supports byte or word operand transfers. The EBI can be configured to support either 8 or 16-bit data bus operation on a per chip-select basis. Dynamic bus sizing by external devices is not supported.

Each port (port referring to the width of the data path that an external device uses during a data transfer) is assigned to particular bits of the data bus. A 16-bit port is assigned to data bus bits [15:0], and an 8-bit port is assigned to data bus bits [15:8].

In multiplexed modes, address and data share the same bus. In the first part of the bus cycle the address is driven on the address/data bus. At this time, an external device should use ALE to latch the address. The address/data bus is then used for data transfer.

Operand bytes are designated as shown in **Figure 12**. OP0 is the most significant byte of a long-word operand, and OP3 is the least significant byte. The two bytes of a word-length operand are OP0 (most significant) and OP1. The single byte of a byte-length operand is OP0.

Operand	Byte Order			
	31 24	23 16	15 8	7 0
Long Word	OP0	OP1	OP2	OP3
Word			OP0	OP1
Byte				OP0

Figure 12 Operand Byte Order

3.5.12 Operand Alignment

The data multiplexer establishes the necessary connections for different combinations of address and data sizes. Positioning of bytes is determined by the SIZE and ADDR0 outputs. SIZE indicates whether a byte (SIZE = 1) or a word (SIZE = 0) remains to be transferred during the current bus cycle.

ADDR0 also affects the operation of the data multiplexer. During an operand transfer, ADDR[17:1] indicate the word base address of the portion of the operand to be accessed, and ADDR0 indicates the byte offset from the base.

3.5.13 Misaligned Operands

The CPU16 processor uses a basic operand size of 16 bits. An operand is misaligned when it overlaps a word boundary. This is determined by the value of ADDR0. When ADDR0 = 0 (an even address), the address is on a word and byte boundary. When ADDR0 = 1 (an odd address), the address is on a byte boundary only. A byte operand is aligned at any address; a word or long-word operand is misaligned at an odd address.

The CPU16 can perform misaligned word transfers. The CPU16 treats misaligned long-word transfers as two misaligned word transfers. Misaligned transfers entail a substantial performance penalty.

3.5.14 Operand Transfer Cases

Address and data buses, which can be either multiplexed or non-multiplexed, and various control signals, bring about the transfer of data between the EBI and external devices. Signals issued by the bus master control data movement across the bus within an asynchronous bus structure.

Table 20 is a summary of how operands are aligned for various types of transfers. OPn entries are portions of a requested operand that are read or written during a bus cycle and are defined by SIZE and ADDR0 for that bus cycle.

Table 20 Operand Alignment

Transfer Case[1]	SIZE	ADDR0	DATA[15:8]	DATA[7:0]
Byte to 8-Bit Port (Even/Odd)	X[2]	0/1	OP0	—
Byte to 16-Bit Port (Even)	1	0	OP0	—
Byte to 16-Bit Port (Odd)	1	1	—	OP0
Word to 8-Bit Port (Aligned/Misaligned)	X[2]	0/1	OP0	—
Word to 16-Bit Port (Aligned)	0	0	OP0	OP1
Word to 16-Bit Port (Misaligned)	X	1	—	OP0

1. There is no way to externally differentiate long word transfers from word transfers. Aligned long word transfers are treated as two aligned word transfers. Misaligned long word transfers are treated as two misaligned word transfers.
2. “X” indicates that the state of the signal has no effect upon operand alignment.

Using external logic, the SIZE and ADDR0 signals can be decoded with an active low chip-select to generate active low upper and lower byte chip-selects for a 16-bit memory block composed of two 8-bit devices. **Figure 13** shows a typical circuit.

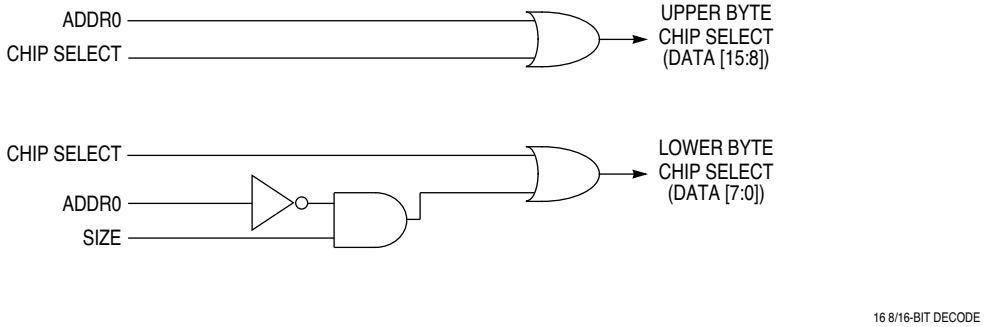


Figure 13 Upper and Lower Byte Chip-Select Generation

3.5.15 External Bus Interface in Low-Power Stop Mode

In low-power stop mode (LPSTOP), the EBI controls the state of the external bus pins, based on the state of the $\overline{EBR}$ pin. If $\overline{EBR}$ is low, the SLIM does not drive the address, address/data, $\overline{AS}$, $\overline{DS}$, R/$\overline{W}$, SIZE, and ALE pins. If $\overline{EBR}$ is high, the SLIM drives the address, address/data, $\overline{AS}$, $\overline{DS}$, R/$\overline{W}$, and SIZE pins to their last known state. $\overline{EBR}$ does not need to be held low during the duration of LPSTOP. Only a short pulse (25 ns) is required.

3.6 Resets

Reset procedures handle system initialization and recovery from catastrophic failure. This MCU performs resets with a combination of hardware and software. The SLIM determines whether a reset is valid, asserts control signals, performs basic system configuration and boot ROM selection based on hardware mode-select inputs, then passes control to the CPU16.

Reset occurs when an active low logic level on the $\overline{RESET}$ pin is clocked into the SLIM. Resets are gated by the CLKOUT signal. Asynchronous resets are assumed to be catastrophic. An asynchronous reset can occur on any clock edge. Synchronous resets are timed to occur at the end of bus cycles. If there is no clock when $\overline{RESET}$ is asserted, reset does not occur until the clock starts. Resets are clocked to allow completion of write cycles in progress at the time $\overline{RESET}$ is asserted.

Reset is the highest-priority CPU16 exception. Any processing in progress is aborted by the reset exception, and cannot be restarted. Only essential tasks are performed during reset exception processing. Other initialization tasks must be accomplished by the exception handler routine.

RSR — Reset Status Register **$YFFA07**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NOT USED								EXT	POW	SW	HLT	0	LOC	SYS	TST
RESET:								CURRENT STATE OF CORRESPONDING PINS							

The reset status register contains a bit for each reset source in the MCU. A bit set to one indicates what type of reset has occurred. Only one bit in RSR will be set at any time. The reset status register is updated by the reset control logic when the MCU comes out of reset. This register can be read at any time. A write has no effect.

EXT — External Reset
Reset was caused by an external signal.

POW — Power-Up Reset
Reset was caused by the power-up reset circuit.

SW — Software Watchdog Reset
Reset was caused by the software watchdog circuit.

HLT — Halt Monitor Reset
Reset was caused by the halt monitor.

LOC — Loss of Clock Reset
Reset was caused by loss of clock submodule frequency reference. This reset can only occur if the RSTEN bit in the clock submodule is set and the VCO is enabled.

SYS — System Reset
Reset was caused by a CPU RESET instruction. The CPU16 has no RESET instruction, therefore, this bit is not used on the MC68HC16V1 and always reads zero.

TST — Test Submodule Reset
Reset was caused by the test submodule.

3.6.1 Background Debug Mode Operation Out of Reset

Unlike M68HC16 family derivatives using a system integration module (SIM) or single-chip integration module (SCIM) that dedicate specific pins to support CPU16 background debug mode (BDM) operation, the SLIM allows the DSO, DSI, $\overline{\text{BKPT}}$/DSCLK, and FREEZE signals to also serve as port D general purpose I/O lines (PD[3:0]). The $\overline{\text{BKPT}}$/DSCLK signal must be driven appropriately during reset to configure port D for BDM operation. **Table 21** shows the configuration port D BDM signals during reset.

Table 21 Port D Reset Configuration

$\overline{\text{BKPT}}$/DSCLK	Port D Configuration
0	PD[3:0] provide background debug mode lines
1	PD[3:0] provide general-purpose I/O lines

3.6.2 Default Configuration Override During Reset

Default reset values for some bits in the PCON and MCRC shadow registers can be overridden by driving specific pins to predefined logic levels during reset when the $\overline{\text{DRCD}}$ pin is driven to logic 0. The following paragraphs describe the different default configuration override options.

3.6.2.1 MCU Operating Mode

The default MCU operating mode can be overridden during reset by driving the $\overline{\text{DS}}$, SIZE, and R/$\overline{\text{W}}$ pins to the states shown in **Table 22**.

Table 22 Default Operating Mode Override

Shadow Bits	$\overline{\text{DS}}$	SIZE	R/$\overline{\text{W}}$	Operating Mode
MCRC[11:9]	0	X	X	Reserved
	1	0	0	Expanded non-multiplexed
	1	0	1	Expanded multiplexed
	1	1	X	Single chip

3.6.2.2 Port C Configuration

The default configuration of PC[1:0] and ADDR[17:16] can be overridden by driving the ADDR/DATA[3:1] pins to the states shown in **Table 23**.

Table 23 Port C Default Configuration Override

Shadow Bit	ADDR3/DATA3	ADDR2/DATA2	ADDR1/DATA1	Pin Configuration
PCON[13:11]	0	0	0	PC[1:0]
	0	0	1	PC1, ADDR16
	0	1	0	ADDR[17:16]
	0	1	1	Reserved
	1	X	X	Reserved

3.6.2.3 Boot ROM Operation

The default configuration of the SLIM boot ROM and the $\overline{CSB}$ bootstrap chip-select can be overridden by driving the ADDR11/DATA11 pin to the states shown in **Table 24**.

Table 24 Default Boot ROM Override

Shadow Bit	ADDR11/DATA11	SLIM Boot ROM	$\overline{CSB}$ Boot Chip-Select
MCR4	0	Enabled	Disabled
	1	Disabled	Enabled

3.6.2.4 Boot Chip-Select Operation

The default operation of $\overline{CSB}$/PD6 and the reset state of the chip-select control register (CSCR) $\overline{CSB}$ data bus width bit (SIZB) can be overridden by driving the ADDR/DATA[7:6] pins to the states shown in **Table 25**.

Table 25 Default Boot Chip-Select Operation Override

Shadow Bit	ADDR7/DATA7	ADDR6/DATA6	Pin Configuration	$\overline{CSB}$ Port Size
PCON[8:7]	0	0	PD6	8-bit
	0	1	PD6	16-bit
	1	0	$\overline{CSB}$	8-bit
	1	1	$\overline{CSB}$	16-bit

3.6.2.5 Clock Mode Selection

The default clock generation mode of the MCU can be overridden by driving the V_{DDSYN} and FREEZE pins to the states shown in **Table 26**.

Table 26 Default MCU Clock Source Override

Shadow Bit[1]	V_{DDSYN}	FREEZE	Clock Generation Mode
PCON0	0	X	External clock mode
	1	0	Fast reference mode
	1	1	Slow reference mode

1. When V_{DDSYN} is driven to logic 0, the MCU operates in external clock mode, regardless of the state of PCON0 or FREEZE.

3.6.2.6 Fast Reference Mode SYNCR Bit Settings

The default settings of the SYNCR X and W[2:0] bits in fast reference mode can be overridden by driving the $\overline{IRQX}$/$\overline{BERR}$ pin to the states shown in **Table 27**.

Table 27 Fast Reference Mode Default SYNCR Configuration Override

Shadow Bit	$\overline{IRQX}$/$\overline{BERR}$	SYNCR X Field	SYNCR W[2:0] Field
PCON2	0	1	000
	1	0	011

3.6.2.7 DTACK Operation

The default operation of DTACK/PD4 can be overridden by driving the ADDR8/DATA8 pin to the states shown in **Table 28**.

Table 28 DTACK/PD4 Default Configuration Override

Shadow Bit	ADDR8/DATA8	Pin Configuration
PCON4	0	PD4
	1	DTACK

3.6.2.8 CPU Function Code Operation

The default operation of FC[2:0]/PE[7:5] can be overridden by driving the ADDR9/DATA9 pin to the states shown in **Table 29**.

Table 29 FC[2:0]/PE[7:5] Default Configuration Override

Shadow Bit	ADDR9/DATA9	Pin Configuration
PCON3	0	PE[7:5]
	1	FC[2:0]

3.6.2.9 CLKOUT Operation

The default operation of CLKOUT/PE4 can be overridden by driving the ADDR0/DATA0 pin to the states shown in **Table 30**.

Table 30 CLKOUT/PE4 Default Configuration Override

Shadow Bit	ADDR0/DATA0	Pin Configuration
PCON1	0	PE4
	1	CLKOUT

3.6.3 MCU Module Pin Function During Reset

Generally, module pins default to port functions, and input/output ports are set to input state. This is accomplished by disabling pin functions in the appropriate control registers, and by clearing the appropriate port data direction registers. Refer to individual module sections in this manual for more information.

3.6.4 Reset Timing

The RESET input must be asserted for a specified minimum period for reset to occur. External RESET assertion can be delayed internally for a period equal to the longest bus cycle time (or the bus monitor time-out period) in order to protect write cycles from being aborted by reset. While RESET is asserted, SLIM pins are either in an inactive, high-impedance state or are driven to their inactive states.

When an external device asserts RESET for the proper period, reset control logic clocks the signal into an internal latch. The control logic drives the RESET pin low for an additional 512 CLKOUT cycles after it detects that the RESET signal is no longer being externally driven, to guarantee this length of reset to the entire system.

If an internal source asserts a reset signal, the reset control logic asserts RESET for a minimum of 512 cycles. If the reset signal is still asserted at the end of 512 cycles, the control logic continues to assert RESET until the internal reset signal is negated.

After 512 cycles have elapsed, the reset input pin goes to an inactive, high-impedance state for 10 cycles. At the end of this 10-cycle period, the $\overline{\text{RESET}}$ input is tested. When the input is at logic level one, reset exception processing begins. If, however, the $\overline{\text{RESET}}$ input is at logic level zero, the reset control logic drives the pin low for another 512 cycles. At the end of this period, the pin again goes to high-impedance state for 10 cycles, then it is tested again. The process repeats until $\overline{\text{RESET}}$ is released.

3.6.5 Power-On Reset

When the SLIM clock synthesizer is used to generate system clocks, power-on reset involves special circumstances related to application of system and clock synthesizer power. When fast or slow reference mode is selected, voltage must be applied to the clock synthesizer power input pin (V_{DDSYN}) for the MCU to operate. The following discussion assumes that, to minimize crystal start-up time, V_{DDSYN} is applied before and during reset. When V_{DDSYN} is applied at power-on, start-up time is affected by specific crystal parameters and by oscillator circuit design. V_{DD} ramp-up time also affects pin state during reset.

During power-on reset, an internal circuit in the SLIM drives the IMB internal and external $\overline{\text{RESET}}$ lines. The circuit releases the internal $\overline{\text{RESET}}$ line as V_{DD} ramps up to the minimum specified value, and SLIM pins are initialized. When V_{DD} reaches minimum value, the clock synthesizer VCO begins operation, and clock frequency ramps up to limp mode frequency. The external $\overline{\text{RESET}}$ signal remains asserted until the clock synthesizer PLL locks and 512 CLKOUT cycles elapse.

The SLIM clock synthesizer provides clock signals to the other MCU modules. After the clock is running and the internal $\overline{\text{RESET}}$ signal is asserted for four clock cycles, these modules reset. V_{DD} ramp time and VCO frequency ramp time determine how long the four cycles take. Worst case is approximately 15 milliseconds. During this period, module port pins may be in an indeterminate state. While input-only pins can be put in a known state by means of external pull-up resistors, external logic on input/output or output-only pins must condition the lines during this time. Active drivers require high-impedance buffers or isolation resistors to prevent conflict.

3.6.6 Use of Three-State Control

When the three state control ($\overline{\text{TSC}}$) input is asserted simultaneously with the $\overline{\text{RESET}}$ pin, the MCU places all output drivers in an inactive, high-impedance state. The signal must remain asserted for 10 clock cycles for drivers to change state.

When using $\overline{\text{TSC}}$ during power-up reset, note the following constraints:

- When the internal PLL clock synthesizer is used, synthesizer ramp-up time affects how long the 10 cycles take. Worst case is approximately 20 milliseconds from $\overline{\text{TSC}}$ assertion.
- When an external clock signal is applied, pins go to a high-impedance state as soon as 10 clock pulses have been applied to the EXTAL pin after $\overline{\text{TSC}}$ assertion.

NOTE

When $\overline{\text{TSC}}$ assertion takes effect, internal signals are forced to values that can cause inadvertent mode selection. Once the output drivers change state, the MCU must be powered down and restarted before normal operation can resume.

3.7 Interrupts

Interrupt recognition and servicing involve complex interaction between the central processing unit, the SLIM, and a device or module requesting interrupt service.

The CPU16 provides for seven levels of interrupt priority (1–7), seven automatic interrupt vectors, and 200 assignable interrupt vectors. All interrupts with priorities less than seven can be masked by the interrupt priority (IP) field in the condition code register. The CPU16 handles interrupts as a type of asynchronous exception.

Interrupt recognition is based on the states of the $\overline{IRQ7}$, $\overline{IRQ2}$, $\overline{IRQX}$, and the IP mask value set in the CPU16 condition code register (CCR). Each of the signals corresponds to an interrupt priority. $\overline{IRQ2}$ has the lowest priority, $\overline{IRQ7}$ has the highest priority. $\overline{IRQX}$ has a user-definable priority based on the setting of the $\overline{IRQXL}$[2:0] bits in PFLVR. Refer to **3.8.5 Port F Operation** for more information on the port F interrupt level register (PFLVR).

The IP field consists of three bits (CCR[7:5]). Binary values %000 to %111 provide eight priority masks. Masks prevent an interrupt request of a priority less than or equal to the mask value (except for $\overline{IRQ7}$) from being recognized and processed. When IP contains %000, no interrupt is masked. During exception processing, the IP field is set to the priority of the interrupt being serviced.

Interrupt request signals can be asserted by external devices or by microcontroller modules. Request lines are connected internally by means of a wired NOR. Simultaneous requests of differing priority can be made. Internal assertion of an interrupt request signal does not affect the logic state of the corresponding MCU pin.

External interrupt requests are routed to the CPU16 via the external bus interface and SLIM interrupt control logic. The CPU treats external interrupt requests as though they come from the SLIM.

$\overline{IRQ7}$, $\overline{IRQ2}$, and $\overline{IRQX}$ can be configured as active-low level-sensitive or edge detect inputs. They may also be configured as general-purpose I/O pins, or edge detect pins, in which case they exist but do not support the $\overline{IRQ}$ function. Control bits in the port F pin assignment register are used to assign the function. Refer to **3.8.5 Port F Operation** for more information.

$\overline{IRQ7}$ is a non-maskable interrupt that is input transition sensitive in order to prevent redundant servicing and stack overflow. A non-maskable interrupt is generated each time $\overline{IRQ7}$ is asserted, and each time the priority mask changes from %111 to a lower number while $\overline{IRQ7}$ is asserted.

$\overline{IRQ2}$ and $\overline{IRQX}$ are maskable interrupts. $\overline{IRQ2}$ asserts fixed interrupt priority level 2. $\overline{IRQX}$ can assert any interrupt priority level as configured in the port F priority level register (PFLVR). Refer to **3.8.5 Port F Operation** for more information.

Interrupt requests are sampled on consecutive falling edges of the system clock. Interrupt request input circuitry has hysteresis. To be valid, a request signal must be asserted for at least two consecutive clock periods. Valid requests do not cause immediate exception processing, but are left pending. Pending requests are processed at instruction boundaries or when exception processing of higher-priority exceptions is complete.

The CPU16 does not latch the priority of a pending interrupt request. If an interrupt source of higher priority makes a service request while a lower priority request is pending, the higher priority request is serviced. If an interrupt request of equal or lower priority than the current IP mask value is made, the CPU does not recognize the occurrence of the request in any way.

3.7.1 Interrupt Acknowledge Cycle

Interrupt acknowledge bus cycles are generated during exception processing. When the CPU16 detects one or more interrupt requests of a priority higher than the interrupt priority mask value, it performs a CPU space read from address $FFFFF : [IP] : 1.

The CPU space read cycle performs two functions: it places a mask value corresponding to the highest priority interrupt request on the address bus, and it acquires an exception vector number from the interrupt source. The mask value also serves two purposes: it is latched into the CCR IP field in order to mask lower-priority interrupts during exception processing, and it is decoded by modules that have requested interrupt service to determine whether the current interrupt acknowledge cycle pertains to them.

Modules that have requested interrupt service decode the IP value placed on the address bus at the beginning of the interrupt acknowledge cycle, and if their requests are at the specified IP level, respond to the cycle. Arbitration between simultaneous requests of the same priority is performed by means of serial contention between module interrupt arbitration (IARB) field values.

Each module that can make an interrupt service request, including the SLIM, has an IARB field in its module configuration register. An IARB field can be assigned a value from %0001 (lowest priority) to %1111 (highest priority). A value of %0000 in an IARB field causes the CPU16 to process a spurious interrupt exception when an interrupt from that module is recognized.

Because the EBI manages external interrupt requests, the SLIM IARB value is used for arbitration between internal and external interrupt requests. The reset value of IARB for the SLIM is %1111, and the reset IARB value for all other modules is %0000. Initialization software must assign different IARB values to implement an arbitration scheme.

Each module must have a unique IARB value. When two or more IARB fields have the same non-zero value, the CPU16 interprets multiple vector numbers simultaneously, with unpredictable consequences.

Arbitration must always take place, even when a single source requests service. This point is important for two reasons: the CPU interrupt acknowledge cycle is not driven on the external bus unless the SLIM wins contention, and failure to contend causes an interrupt acknowledge bus cycle to be terminated by a bus error, which causes a spurious interrupt exception to be taken.

When arbitration is complete, the dominant module must place an interrupt vector number on the data bus and terminate the bus cycle. In the case of an external interrupt request, because the interrupt acknowledge cycle is transferred to the external bus, an external device must decode the mask value and respond with a vector number, then generate bus cycle termination signals. If the device does not respond in time, a spurious interrupt exception is taken.

3.7.2 Interrupt Processing Summary

A summary of the interrupt processing sequence follows. When the sequence begins, a valid interrupt service request has been detected and is pending.

A. The CPU finishes higher priority exception processing or reaches an instruction boundary.
B. Processor state is stacked, then the CCR PK extension field is cleared.
C. The interrupt acknowledge cycle begins:
    1. FC[2:0] are driven to %111 (CPU space) encoding.
    2. The address bus is driven as follows. ADDR[23:20] = %1111; ADDR[19:16] = %1111, which indicates that the cycle is an interrupt acknowledge CPU space cycle; ADDR[15:4] = %111111111111; ADDR[3:1] = the priority of the interrupt request being acknowledged; and ADDR0 = %1.
    3. Request priority is latched into the CCR IP field from the address bus.

D. Modules or external peripherals that have requested interrupt service decode the priority value on ADDR[3:1]. If request priority is the same as the priority value in the address, IARB contention takes place. When there is no contention, the spurious interrupt monitor asserts $\overline{\text{BERR}}$, and a spurious interrupt exception is processed.
E. After arbitration, the interrupt acknowledge cycle can be completed in one of three ways:
    1. The dominant interrupt source supplies a vector number and $\overline{\text{DTACK}}$ to terminate the bus cycle. The CPU16 acquires the vector number.
    2. The internal $\overline{\text{AVEC}}$ signal is asserted by the dominant interrupt source and the CPU16 generates an autovector number corresponding to interrupt priority.
    3. The bus monitor asserts $\overline{\text{BERR}}$ and the CPU16 generates the spurious interrupt vector number.
F. The vector number is converted to a vector address.
G. The content of the vector address is loaded into the PC, and the processor transfers control to the exception handler routine.

3.8 General-Purpose Input/Output

Most of the pins associated with the SLIM can be used for several different functions. Their primary function is to provide an external bus interface for applications that require access to off-chip resources. When not being used for their primary functions, these pins can be used as digital I/O pins. To facilitate I/O functions, the SLIM pins are grouped into 8-bit ports. Each port has associated registers that are used to configure the pins for the desired functions.

The SLIM contains eight general-purpose input/output ports: A, B, C, D, E, F, G, and H. All are available as either input or output. All ports have two associated data registers used to monitor or control the state of its pins.

The port output data register is readable and writable. Writes to the port output data registers cause IMB data to be latched, which is then driven to the pads of all pins programmed as outputs. Reads of the output data returns the current state of the data latch for all pins, regardless of the actual state of the output pin.

The pin data registers return the current state of all pins regardless of whether they are input or output. After reset, all pin data registers reflect the actual state of the pin, as reset causes all pins to become inputs by clearing the data direction registers.

Ports C, D, E, and F have pin assignment registers that show if individual pins are configured for digital I/O, or some other function. Ports A, B, and G do not have pin assignment registers and are configured as digital I/O based on the mode of the SLIM as configured during reset. Port H does not have a pin assignment register and is configured as digital I/O based on the mode of the SLIM and the programming of the data port size bits of the chip selects ($\overline{\text{CSA}}$, $\overline{\text{CSB}}$, and $\overline{\text{CSC}}$).

All ports have an associated data direction register that is used to configure port pins as either outputs or inputs. In addition to the output data and pin data registers, port F has an edge-detect flag register that indicates whether a transition has occurred on any of its pins.

Table 31 shows the primary functions of the general-purpose I/O ports and the modes in which the I/O ports are available. “X” indicates port availability in a particular mode.

Table 31 I/O Port Functions and Availability

Port	Primary Function	Availability			
		Single-Chip	Expanded Non-Multiplexed (16-Bit Data Bus)	Expanded Non-Multiplexed (8-Bit Data Bus)	Expanded Multiplexed
A	ADDR/DATA[15:8]	X	—	—	—
B	ADDR/DATA[7:0]	X	—	—	—
C	ADDR[17:16]	X	X	X	X
D	Chip-selects, background debug mode	X	X	X	X
E	Bus control	X	X	X	X
F	Interrupts, bus control	X	X	X	X
G	DATA[15:8]	X	—	—	X
H	DATA[7:0]	X	—	X	X

3.8.1 Port A and B Operation

Ports A and B can be configured as either input or output ports on a whole port basis. Depending on the mode of the SLIM configured during reset, port A and port B can come out of reset as digital inputs or as address bus pins.

PORTA — Port A Output Data Register **$YFFA10**
PORTB — Port B Output Data Register **$YFFA11**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0
RESET:															
U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U

Port A and B output data registers are used to latch the data to be driven on their respective port output pins if port A or port B is configured as an output. When read, ports A and B output data registers always reflect the current state of the data latches. Power-on reset can change the state of these latches. All other sources of reset have no effect.

PORTAP — Port A Pin Data Register **$YFFA12**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PAP7	PAP6	PAP5	PAP4	PAP3	PAP2	PAP1	PAP0	PORTBP							
RESET:															
CURRENT STATE OF CORRESPONDING PINS															

PORTBP — Port B Pin Data Register **$YFFA13**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PORTAP								PBP7	PBP6	PBP5	PBP4	PBP3	PBP2	PBP1	PBP0
RESET:															
CURRENT STATE OF CORRESPONDING PINS															

When read, pin data registers A and B reflect the current state of their respective pins, regardless of whether the pins are configured as inputs or outputs. Writes have no effect.

DDRAB — Port A/B Data Direction Register **$YFFA15**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NOT USED														DDA	DDB
RESET (SINGLE-CHIP MODE):															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Port A and B data direction registers control the direction of the pin drivers for ports A and B, respectively, when the pins are configured for I/O. Setting DDA or DDB to one configures all pins in the corresponding port as outputs. Clearing DDA or DDB to zero configures all pins in the corresponding port as inputs.

3.8.2 Port C Operations

Port C pins can be used as address lines or general purpose I/O. ADDR[17:16] are selected in a single contiguous left-justified group, to form a contiguous address space with DATA[15:0].

PORTC — Port C Output Data Register **$YFFA18**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED						PC1	PC0	PORTD							
RESET:															
U	U	U	U	U	U	U	U								

The port C output data register latches the data to be driven on the port C output pins. When read, PORTC always reflects the current state of the data latches. Power-on reset can change the state of these latches. Other sources of reset have no effect.

PORTCP — Port C Pin Data Register **$YFFA1A**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED						PCP1	PCP0	PORTDP							
RESET:															
CURRENT STATE OF CORRESPONDING PINS															

When read, the port C pin data register reflects the current state of the port C pins, regardless of whether the pins are configured as inputs or outputs. Writes to PORTCP have no effect.

DDRC — Port C Data Direction Register **$YFFA1C**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED						DDC1	DDC0	DDRD							
RESET:															
0	0	0	0	0	0	0	0								

Port C data direction register bits control the direction of the port C pin drivers when the pins are configured as I/O pins. When any bit in this register is set to one, the corresponding pin is configured as an output. When any bit in this register is cleared to zero, the corresponding pin is configured as an input. Reset clears all bits to zero.

PCPAR — Port C Pin Assignment Register **$YFFA1E**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NOT USED					PCPA[2:0]			PDPAR							
RESET (EXPANDED MODE):															
0	0	0	0	0	PCON 13	PCON 12	PCON 11								
RESET (SINGLE-CHIP MODE):															
0	0	0	0	0	0	0	0								

The port C pin assignment register controls the function of all port C pins.

In expanded modes, the pin assignment register resets to the default reset states programmed in the port/clock configuration shadow register (PCON[13:11]).

PC[1:0] can be configured as address pins (ADDR[17:16]) or digital I/O. Refer to **Table 32**.

Table 32 PC[1:0] Pin Assignment Encoding

PCPAR[10:8]	Port C I/O Pins	Address Pins
0 0 0	PC[1:0]	None
0 0 1	PC1	ADDR16
0 1 0	None	ADDR[17:16]
0 1 1	Reserved	Reserved
1 X X	Reserved	Reserved

3.8.3 Port D Operations

Port D pins can be individually selected as either digital I/O or external bus control and debug signals, depending on the mode of the SLIM configured during reset.

PORTD — Port D Output Data Register **$YFFA19**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PORTC								PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0
RESET:															
								U	U	U	U	U	U	U	U

The port D output data register latches the data to be driven on the port D output pins. When read, PORTD always reflects the current state of the data latches. Power-on reset can change the state of these latches. All other sources of reset have no effect.

PORTDP — Port D Pin Data Register **$YFFA1B**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PORTC								PDP7	PDP6	PDP5	PDP4	PDP3	PDP2	PDP1	PDP0
RESET:															
CURRENT STATE OF CORRESPONDING PINS															

When read, the port D pin data register reflects the current state of the port D pins, regardless of whether the pins are configured as inputs or outputs. Writes have no effect.

DDRD — Port D Data Direction Register **$YFFA1D**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DDRC								DD7	DD6	DD5	DD4	DD3	DD2	DD1	DD0
RESET:								0	0	0	0	0	0	0	0

Port D data direction register bits control the direction of the port D pin drivers when the pins are configured as I/O pins. When any bit in this register is set to one, the corresponding pin is configured as an output. When any bit in this register is cleared to zero, the corresponding pin is configured as an input. Reset clears all bits to zero.

PDPAR — Port D Pin Assignment Register **$YFFA1F**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PCPAR								PDPA7	PDPA6	PDPA5	PDPA4	PDPA3	PDPA2	PDPA1	PDPA0
RESET (EXPANDED MODE):								PCON10	PCON8	PCON6	PCON4	$\overline{\text{BKPT}}$	$\overline{\text{BKPT}}$	$\overline{\text{BKPT}}$	$\overline{\text{BKPT}}$
RESET (SINGLE-CHIP MODE):								0	0	0	0	$\overline{\text{BKPT}}$	$\overline{\text{BKPT}}$	$\overline{\text{BKPT}}$	$\overline{\text{BKPT}}$

Port D pin assignment register bits control the function of each port D pin. Refer to **Table 33**. When any bit in this register is set to one, the corresponding pin is configured as its primary function. When any bit in this register is cleared to zero, the corresponding pin is configured as an I/O pin controlled by the port D output data and data direction registers.

The SLIM mode configured during reset and the state of bits in the port/clock configuration shadow register are used to configure PDPAR[7:4] during reset in expanded modes. Shadow register configuration can be overridden by external pins during reset. PDPAR[3:0] are always reset to the inverted state of the $\overline{\text{BKPT}}$ pin.

Table 33 Port D Pin Assignments

PDPAR Bit	Port D Signal	Primary Function
PDPA7	PD7	$\overline{\text{CSA}}$
PDPA6	PD6	$\overline{\text{CSB}}$
PDPA5	PD5	$\overline{\text{CSC}}$
PDPA4	PD4	$\overline{\text{DTACK}}$
PDPA3	PD3	FREEZE
PDPA2	PD2	$\overline{\text{BKPT}}$
PDPA1	PD1	IPIPE1
PDPA0	PD0	IPIPE0

3.8.4 Port E Operations

Port E pins can be individually selected as either digital I/O or bus control signals depending on the SLIM mode configured during reset.

PORTE — Port E Output Data Register **$YFFA21**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NOT USED								PE7	PE6	PE5	PE4	PE3	PE2	PE1	PE0
RESET:								U	U	U	U	U	U	U	U

The port E output data register latches the data to be driven on the port E output pins. When read, this register always reflects the current state of the data latches. Power-on reset can change the state of these latches. All other sources of reset have no effect.

PORTEP — Port E Pin Data Register **$YFFA23**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NOT USED								PEP7	PEP6	PEP5	PEP4	PEP3	PEP2	PEP1	PEP0
RESET:															
CURRENT STATE OF CORRESPONDING PINS															

When read, the port E pin data register reflects the current state of the port E pins, regardless of whether the pins are configured as inputs or outputs. Writes have no effect.

DDRE — Port E Data Direction Register **$YFFA25**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NOT USED								DDE7	DDE6	DDE5	DDE4	DDE3	DDE2	DDE1	DDE0
RESET:								0	0	0	0	0	0	0	0

Port E data direction register bits control the direction of the port E pin drivers when the pins are configured as I/O pins. When any bit in this register is set to one, the corresponding pin is configured as an output. When any bit in this register is cleared to zero, the corresponding pin is configured as an input. Reset clears all bits to zero.

PEPAR — Port E Pin Assignment Register **$YFFA27**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NOT USED								PEPA7	PEPA6	PEPA5	PEPA4	PEPA3	PEPA2	PEPA1	PEPA0
RESET (EXPANDED MODE):								PCON3	PCON3	PCON3	PCON1	1	1	1	1
RESET (SINGLE-CHIP MODE):								0	0	0	PCON1	0	0	0	0

Port E pin assignment register bits control the function of each port E pin. Refer to **Table 34**. When any bit in this register is set to one, the corresponding pin is configured as its primary function. When any bit in this register is cleared to zero, the corresponding pin is configured as an I/O pin controlled by the port E output data and data direction registers.

The SLIM mode configured during reset and the state of bits in the port/clock configuration shadow register are used to configure PEPAR during reset. Shadow register configuration can be overridden by external pins during reset.

Table 34 Port E Pin Assignments

PEPAR Bit	Port E Signal	Primary Function
PEPA7	PE7	FC2
PEPA6	PE6	FC1
PEPA5	PE5	FC0
PEPA4	PE4	CLKOUT
PEPA3	PE3	SIZE
PEPA2	PE2	$\overline{AS}$
PEPA1	PE1	$\overline{DS}$
PEPA0	PE0	R/$\overline{W}$

3.8.5 Port F Operation

Port F pins can be individually selected as either digital I/O or bus control signals and interrupt request lines, depending on the SLIM mode configured during reset.

$\overline{IRQ7}$/PF7 and $\overline{IRQ2}$/PF2 can be programmed to operate as either level-sensitive interrupt request inputs, edge-sensitive interrupt request inputs, rising edge-detect I/O pins or falling edge-detect I/O pins. $\overline{IRQX}$/PF1 can be programmed to operate as either the bus error signal ($\overline{BERR}$), a level-sensitive or edge-sensitive interrupt request input of priority 1 through 7, a rising edge-detect I/O pin, or a falling edge-detect I/O pin. ALE/PF0 can be programmed to operate as either the address latch enable signal (ALE), a rising edge-detect I/O pin or a falling edge detect I/O pin.

Table 35 summarizes port F pin operation.

Table 35 Port F Operation

Pin	Pin Function
$\overline{IRQ7}$, $\overline{IRQ2}$	Fixed priority, level-sensitive interrupt input
	Edge-sensitive interrupt input
	Rising or falling edge-detect, digital I/O
$\overline{IRQX}$	Programmable priority level-sensitive interrupt input
	Edge-sensitive interrupt input
	Rising or falling edge-detect, digital I/O
ALE	Address latch enable, rising or falling edge-detect, digital I/O

PORTF — Port F Output Data Register **$YFFA31**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NOT USED								PF7	RESERVED				PF2	PF1	PF0
RESET:								U	U	U	U	U	U	U	U

The port F output data register latches the data to be driven on the port F output pins. When read, this register always reflects the current state of the data latches. Power-on reset can change the state of these latches. All other sources of reset have no effect.

PORTFP — Port F Pin Data Register **$YFFA33**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NOT USED								PFP7	RESERVED				PFP2	PFP1	PFP0
RESET:															
CURRENT STATE OF CORRESPONDING PINS															

When read, the port F pin data register reflects the current state of the port F pins, regardless of whether the pins are configured as inputs or outputs. Writes have no effect.

DDRF — Port F Data Direction Register **$YFFA35**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NOT USED								DDF7	RESERVED				DDF2	DDF1	DDF0
RESET:															
								0	0	0	0	0	0	0	0

Port F data direction register bits control the direction of the port F pin drivers when the pins are configured as I/O pins. When any bit in this register is set to one, the corresponding pin is configured as an output. When any bit in this register is cleared to zero, the corresponding pin is configured as an input. Reset clears all bits to zero.

PFPAR — Port F Pin Assignment Register **$YFFA36**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PFPA7		RESERVED								PFPA2		PFPA1		PFPA0	
RESET															
1	1	1	1	1	1	1	1	1	1	1	1	1	1	*	*

*0 if multiplexed SLIM configuration is selected, otherwise 1

Each of the eight fields in this register controls the function of a port F pin. Field encodings 01 and 10 configure the pin as a rising or falling edge-detect I/O pin controlled by the port F data and data direction registers.

When a field in this register is set to 11, the corresponding pin is configured as a level-sensitive interrupt request input for pins $\overline{IRQ7}$/PF7, $\overline{IRQ2}$/PF2, and $\overline{IRQX}$/PF1, and as digital I/O on ALE/PF0.

When a field in this register is programmed to 00, the corresponding pin is configured as an edge-sensitive interrupt request input for pins $\overline{IRQ7}$/PF7, $\overline{IRQ2}$/PF2, and $\overline{IRQX}$/PF1, and as digital I/O on pin ALE/PF0.

The $\overline{IRQX}$/$\overline{BERR}$ signal operates as bus error when $\overline{IRQXL[2:0]}$ = %000 in the port F interrupt level register (PFLVR), regardless of the setting of the port F pin assignment register PFPA1 field.

Refer to **Table 36** for port F pin assignments. **Table 37** displays port F pin assignment register field encodings.

Table 36 Port F Pin Assignments

PFPAR FIELD	Port F Signal	Primary Function
PFPA7	PF7	$\overline{\text{IRQ7}}$
PFPA6	Reserved	Reserved
PFPA5	Reserved	Reserved
PFPA4	Reserved	Reserved
PFPA3	Reserved	Reserved
PFPA2	PF2	$\overline{\text{IRQ2}}$
PFPA1	PF1	$\overline{\text{IRQX}}$/$\overline{\text{BERR}}$
PFPA0	PF0	ALE

Table 37 Port F Pin Assignment Register Field Encodings

MSB	LSB	Description
0	0	Edge-sensitive interrupt request input (I/O on PF0)
0	1	Rising edge-detect I/O pin
1	0	Falling edge-detect I/O pin
1	1	Level-sensitive interrupt request (I/O on PF0)

PORTFE — Port F Edge-Detect Flag Register **$YFFA39**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NOT USED								PFE7	RESERVED				PFE2	PFE1	PFE0
RESET:								0	0	0	0	0	0	0	0

The port F edge detect flag register indicates when the proper transition has occurred on a port F pin programmed for edge-detect I/O. A flag is set in this register any time the specified edge is detected on a corresponding pin, regardless of whether the pin is configured as an input or output. If a pin is configured as an interrupt request, bus error ($\overline{\text{BERR}}$), or address latch enable (ALE), the flag for that pin is not set on a transition. To clear flags in PORTFE, read the register and then write to the register with zeros in the positions of the flags to be cleared.

PFEER — Port F Edge-Detect I/O Interrupt Enable Register **$YFFA3B**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NOT USED								PFEE7	RESERVED				PFEE2	PFEE1	PFEE0
RESET:								0	0	0	0	0	0	0	0

Port F edge-detect I/O interrupt enable register bits are used to enable interrupts from individual edge-detect I/O pins. If a bit is set to one, the specified edge on the corresponding pin generates an interrupt request at the interrupt priority level specified by the PFEL[2:0] field of the port F interrupt level register. If the bit is zero, the corresponding edge-detect I/O pin does not generate an interrupt request when an edge is detected. In either case, the corresponding edge detect flag is set in the port F edge-detect flag register.

PFLVR — Port F Interrupt Level Register **$YFFA3C**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	$\overline{\text{IRQXL[2:0]}}$			0	PFEL[2:0]			PFIVR							
RESET:															
0	0	0	1	0	0	0	0								

The port F interrupt level register controls the interrupt levels requested by the $\overline{\text{IRQX}}$ programmable priority interrupt and the port F edge-detect I/O interrupts. Reads of PFLVR7 and PFLVR3 always return zero; writes have not effect.

$\overline{\text{IRQXL[2:0]}}$ — $\overline{\text{IRQX}}$ Level

This field determines the interrupt level and function of the $\overline{\text{IRQX}}$/$\overline{\text{BERR}}$ pin. When $\overline{\text{IRQXL[2:0]}}$ = %000, the $\overline{\text{IRQX}}$/$\overline{\text{BERR}}$ pin functions as the bus error input. When set to any non-zero value, the pin functions as an interrupt request with the specified priority level. $\overline{\text{IRQX}}$/$\overline{\text{BERR}}$ comes out of reset configured to function as an interrupt request of priority level 1.

PFEL[2:0] — Port F Edge-Detect I/O Interrupt Level

This field specifies the priority level of port F edge-detect I/O interrupts; PFEL[2:0] = %000 disables these interrupts. Whenever all SLIM interrupt sources are contending for the same priority level, port F edge-detect interrupts have the lowest arbitration priority.

PFIVR — Port F Edge-Detect I/O Interrupt Vector Register **$YFFA3D**

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

The port F edge-detect I/O interrupt vector register determines which vector in the exception vector table is used for interrupts generated by the port F edge-detect logic.

3.8.6 Port G and H Operation

Ports G and H can be used as input/output ports when the SLIM is configured in expanded multiplexed mode. In addition, port H may be used for digital I/O pin in expanded non-multiplexed mode if all of the chip-select port size bits in the chip-select control register (CSCR) are configured for 8-bit data bus operation.

Ports G and H can be configured as data bus pins or discrete I/O, depending on the mode of the SLIM configured during reset.

PORTG — Port G Output Data Register **$YFFA28**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PG7	PG6	PG5	PG4	PG3	PG2	PG1	PG0	PORTH							
RESET:															
U	U	U	U	U	U	U	U								

The port G output data register latches the data to be driven on the port G output pins. When read, this register always reflects the current state of the data latches. Power-on reset can change the state of these latches. All other sources of reset have no effect.

PORTGP — Port G Pin Data Register **$YFFA2A**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PGP7	PGP6	PGP5	PGP4	PGP3	PGP2	PGP1	PGP0	PORTHP							
RESET:															
CURRENT STATE OF CORRESPONDING PINS															

When read, the port G pin data register reflects the current state of the port G pins, regardless of whether the pins are configured as inputs or outputs. Writes have no effect.

DDRG — Port G Data Direction Register **$YFFA2C**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DDG7	DDG6	DDG5	DDG4	DDG3	DDG2	DDG1	DDG0	DDRH							
RESET:															
0	0	0	0	0	0	0	0								

Port G data direction register bits control the direction of the port G pin drivers when the pins are configured as I/O pins. When any bit in this register is set to one, the corresponding pin is configured as an output. When any bit in this register is cleared to zero, the corresponding pin is configured as an input. Reset clears all bits to zero.

PORTH — Port H Output Data Register **$YFFA29**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PORTG								PH7	PH6	PH5	PH4	PH3	PH2	PH1	PH0
RESET:															
								U	U	U	U	U	U	U	U

The port H output data register latches the data to be driven on the port H output pins. When read, this register always reflects the current state of the data latches. Power-on reset can change the state of these latches. All other sources of reset have no effect.

PORTHP — Port H Pin Data Register **$YFFA2B**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PORTGP								PHP7	PHP6	PHP5	PHP4	PHP3	PHP2	PHP1	PHP0
RESET:															
CURRENT STATE OF CORRESPONDING PINS															

When read, the port H pin data register reflects the current state of the port H pins, regardless of whether the pins are configured as inputs or outputs. Writes have no effect.

DDRH — Port H Data Direction Register **$YFFA2D**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DDRG								DDH7	DDH6	DDH5	DDH4	DDH3	DDH2	DDH1	DDH0
RESET:															
								0	0	0	0	0	0	0	0

Port H data direction register bits control the direction of the port H pin drivers when the pins are configured as I/O pins. When any bit in this register is set to one, the corresponding pin is configured as an output. When any bit in this register is cleared to zero, the corresponding pin is configured as an input. Reset clears all bits to zero.

3.9 Chip-Selects

Typical microcontrollers require additional hardware to provide external chip-select and address decode signals. This MCU provides three independently programmable chip selects, $\overline{CSA}$, $\overline{CSB}$, and $\overline{CSC}$, that provide bus timing and bus cycle termination for external memory and peripheral devices.

Each chip-select has its own base address register (CSBARA, CSBARB, CSBARC) and its own option register (CSORA, CSORB, CSORC). These registers contain the programmable characteristics of a particular chip-select. **Figure 14** displays a block diagram of a programmable chip-select circuit.

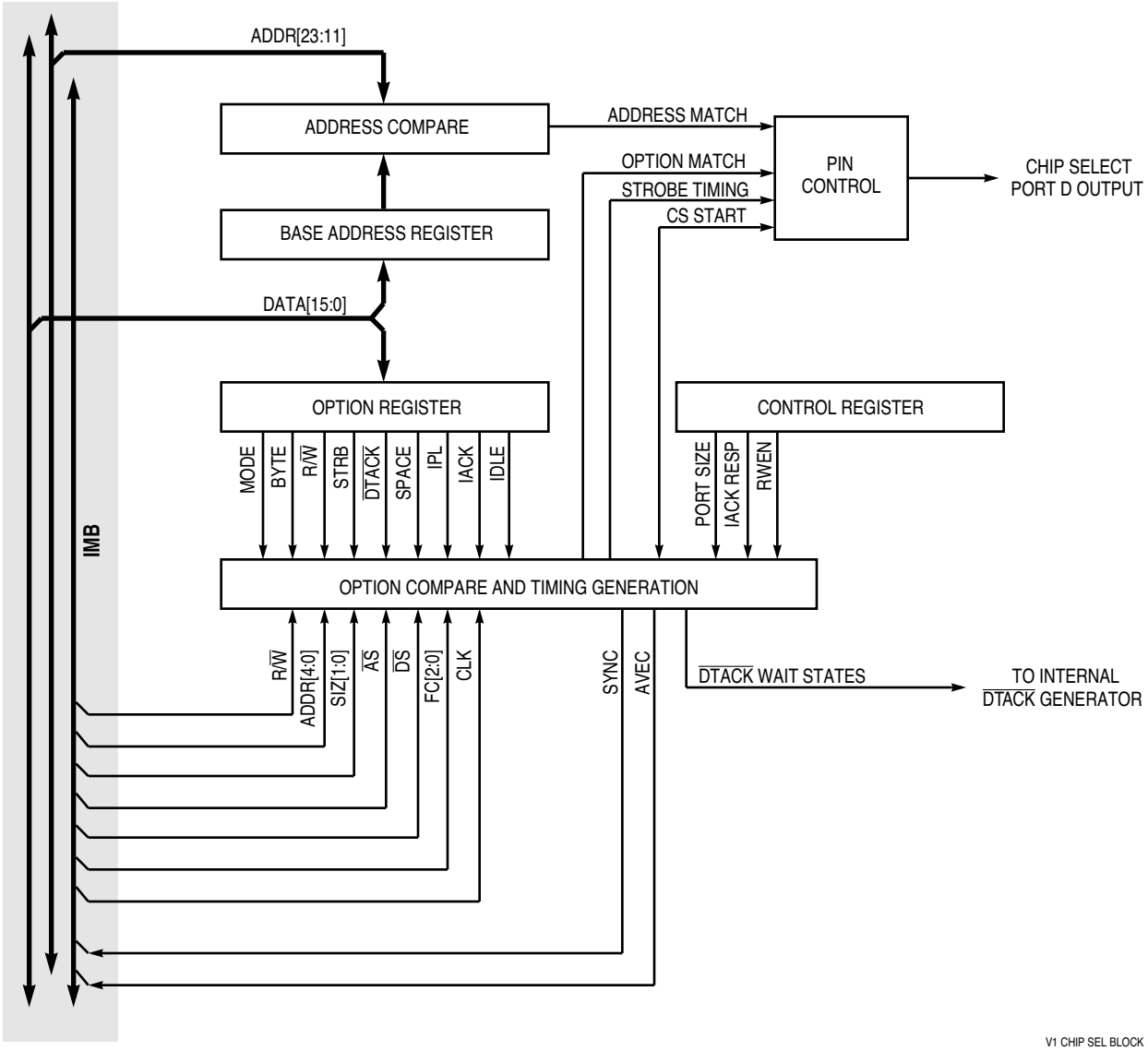


Figure 14 Chip-Select Block Diagram

3.9.1 Programmable Chip-Select Circuit

Chip-select assertion can be synchronized with bus control signals to provide output enable, read/write strobes, or interrupt acknowledge signals. Logic can also generate $\overline{\text{DTACK}}$ signals internally. A single $\overline{\text{DTACK}}$ generator is shared by all circuits. Multiple chip selects assigned to the same block of addresses must have the same number of wait states.

Blocks of addresses are assigned to each chip-select function with sizes of 2 Kbytes to 1 Mbyte. They can be selected by writing values to the appropriate base address register (CSBAR). However, because the logic state of ADDR20 always follows that of ADDR19 on the CPU16, the largest usable block size is 512 Kbytes. Address blocks for separate select functions can overlap.

NOTE

The largest practical block size on the MC68HC16V1 is 256 Kbytes because the ADDR[23:18] pins are not bonded.

When a memory access occurs, chip-select logic compares address space type, address, type of access, transfer size, and interrupt priority (in the case of interrupt acknowledge) to parameters stored in the chip-select registers. If all parameters match, the appropriate chip-select signal is asserted. Select signals are active low.

If a chip-select function is given the same address as a microcontroller module, an access to that address goes to the module, and the chip-select signal is not asserted.

3.9.2 Chip-Select Pin Assignment

Each chip-select pin can have two or more functions. Chip-select configuration out of reset is determined by operating mode. In all expanded modes, the boot ROM select signal ($\overline{\text{CSB}}$) is automatically asserted out of reset if the SLIM boot ROM is disabled. In single-chip mode, all chip select pins are configured as PD[7:5].The data bus size is controlled by the port size programmed in the chip-select control register ($\overline{\text{CSCR}}$). If $\overline{\text{CSA}}$, $\overline{\text{CSB}}$, and $\overline{\text{CSC}}$ are all programmed to 8-bit size, DATA[7:0] becomes port H and DATA[15:8] are used as the 8-bit data bus. If any chip-select is configured with a 16-bit port size, DATA[15:0] are configured as the 16-bit data bus and port H is not available.

NOTE

In expanded mode, the port sizes in the chip-selects must be programmed to determine the data bus size, whether or not chip-select pins are available or are being used.

Three independent bits in CSCR determine the port size for each chip select as 8 or 16 bits. Three other bits in the port D pin assignment register (PDPAR) determine if each chip-select is used as digital I/O or as a chip-select (refer to **Table 38**). The state of these bits at reset is determined by the corresponding mask-programmed shadow bits in the port/clock configuration shadow register (PCON). Default reset values for some bits in the shadow register can be overridden by the state of specific pins during reset. Refer to **3.6.2 Default Configuration Override During Reset** for more information.

Table 38 Chip-Select Pin Assignment Encoding

Chip-Select	PDPAR Bit	PDPAR Shadow Bit	Pin Function	CSCR Bit	CSCR Shadow Bit	Port Size
$\overline{\text{CSA}}$	PDPAR7	PCON10	0 = PD7 1 = $\overline{\text{CSA}}$	SIZA	PCON9	0 = 8 Bit 1 = 16 Bit
$\overline{\text{CSB}}$	PDPAR6	PCON8	0 = PD6 1 = $\overline{\text{CSB}}$	SIZB	PCON7	
$\overline{\text{CSC}}$	PDPAR5	PCON6	0 = PD5 1 = $\overline{\text{CSC}}$	SIZC	PCON5	

3.9.3 Chip-Select Control Register

The chip select control register contains bits controlling the size of accesses made by each of the chip selects, the response to IACK cycles when not programming a chip select to match the cycle, and the option to re-program the R/$\overline{W}$ and $\overline{DS}$ pins as alternate read and write strobes.

CSCR — Chip-Select Control Register **$YFFA6C**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IPL7	IPL6	IPL5	IPL4	IPL3	IPL2	IPL1	IACK	MUXA	MUXB	MUXC	ADRDIS	SIZA	SIZB	SIZC	RWEN
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	PCON9	PCON7	PCON5	0

IPL[7:1] — Interrupt Priority Level

Setting any of the bits in this field to one determines whether or not $\overline{CSA}$ responds to IACK cycles of the corresponding interrupt level. If all bits in IPL[7:1] remain cleared to zero, $\overline{CSA}$ does not respond to IACK cycles. If one or more of these bits is set to one, $\overline{CSA}$ responds according to the IACK bit and the options programmed for $\overline{CSA}$.

IACK — Interrupt Acknowledge Response

This bit, along with the IACK/CS bit in CSORA, determines $\overline{CSA}$ response to an IACK cycle. The response, based on the programming of these two bits, is summarized in **Table 39**.

Table 39 IACK Cycle Response To External $\overline{IRQ}$ Input Assertion

IACK Cycle Conditions				Response			
$\overline{IRQ}$ Pin Asserted	**State of IPL Bit in CSCR Corresponding to $\overline{IRQ}$ Level**	**State of IACK Bit in CSCR**	**State of IACK/CS Bit in CSORA**	**IACK Cycle Termination**	**$\overline{AS}$**	**$\overline{CSA}$**	**Source of Vector**
N	—	—	—	Internal bus error	1	1	None
Y	0	0	0	None/Bus Monitor/ Internal bus error	1	1	None
Y	1	0	0	Internal Autovector	1	1	Corresponding Autovector
Y	—	0	1	External $\overline{DTACK}$	0	1	External
Y	—	1	0	Internal $\overline{DTACK}$[1]	0	1	External
Y	—	1	1	Internal $\overline{DTACK}$[1]	0	0	External

1. Can also be terminated by $\overline{DTACK}$ pin (if available).

MUX[A:C] — Non-Multiplexed Bus Override

These three bits allow multiplexed bus cycles to occur on a non-multiplexed external bus on a per chip-select basis. If a match occurs in a chip-select option and base address register pair, and the MUX bit for the corresponding chip-select is set to one, the access to that block of memory is done using multiplexed address and data.

NOTE

When the SLIM is configured at reset for multiplexed bus operation, these bits have no meaning and are ignored.

ADRDIS — Address Bus Disable

This bit disables the external address bus during internal accesses. When set, the external address bus is not driven on internal cycles unless the show cycle feature is being used. Disabling the address bus results in the addition of one wait state for all cycles, internal and external.

0 = Enable external address bus during internal accesses.
1 = Disable external address bus during internal accesses.

SIZ[A:C] — Chip Select Port Size

These bits determine the port size for each chip select as 8 or 16 bits. The state of these bits at reset is determined by the corresponding mask-programmed shadow bits in the port/clock configuration shadow register (PCON).

0 = 8-bit port.
1 = 16-bit port.

RWEN — Read and Write Strobe Enable

0 = R/$\overline{W}$ and $\overline{DS}$ pins function normally.
1 = R/$\overline{W}$ becomes $\overline{WR}$; $\overline{DS}$ becomes $\overline{RD}$.

3.9.4 Chip-Select Base Address Registers

The base address is the starting address for a block enabled by the given chip select. The block size determines the extent of the address space from its base address. Each chip-select has an associated base address register so that an efficient address map can be constructed for each application.

CSBARA — Chip-Select Base Address Register A **$YFFA60**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR 23*	ADDR 22*	ADDR 21*	ADDR 20*	ADDR 19	ADDR 18	ADDR 17	ADDR 16	ADDR 15	ADDR 14	ADDR 13	ADDR 12	ADDR 11	BLKSZ[2:0]		
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CSBARB — Chip-Select Base Address Register B **$YFFA64**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR 23*	ADDR 22*	ADDR 21*	ADDR 20*	ADDR 19	ADDR 18	ADDR 17	ADDR 16	ADDR 15	ADDR 14	ADDR 13	ADDR 12	ADDR 11	BLKSZ[2:0]		
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1

CSBARB, the option register for $\overline{CSB}$, contains special reset values that support bootstrap operation from peripheral memory devices.

CSBARC — Chip-Select Base Address Register C **$YFFA68**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR 23*	ADDR 22*	ADDR 21*	ADDR 20*	ADDR 19	ADDR 18	ADDR 17	ADDR 16	ADDR 15	ADDR 14	ADDR 13	ADDR 12	ADDR 11	BLKSZ[2:0]		
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

*ADDR[23:20] follow the state of ADDR19 in the MC68HC16V1. Therefore, ADDR[23:20] must match ADDR19 in the chip-select base address registers for the chip-select to be active.

ADDR[23:11] — Base Address Field

This field sets the starting address of a particular address space. The address compare logic uses only the most significant bits to match an address within a block. The value of the base address must be a multiple of block size.

NOTE

Because ADDR[23:20] follow the state of ADDR19 in the CPU16, the maximum block size is 512 Kbytes. For this same reason, addresses from $080000 to $F7FFFF are inaccessible. Blocks can be based above this dead zone, but the effect of ADDR19 must be considered.

BLKSZ[2:0] — Block Size Field

Block size is determined by bits [2:0] of each base address register. This field is used to determine the size of the address space accessed by the chip select. Refer to **Table 40**.

Table 40 Block Size Field Encoding

BLKSZ[2:0]	Block Size (Bytes)	Address Lines Compared
000	2K	ADDR[23:11]
001	8K	ADDR[23:13]
010	16K	ADDR[23:14]
011	64K	ADDR[23:16]
100	128K	ADDR[23:17]
101	256K	ADDR[23:18]
110	512K	ADDR[23:19]
111	512K	ADDR[23:20]

3.9.5 Option Registers

The option registers contain eight fields that determine the timing of and conditions for assertion of chip-select signals.

CSORA — Chip-Select Option Register A **$YFFA62**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MODE	BYTE[1:0]		R/$\overline{W}$[1:0]		STRB	$\overline{DTACK}$[3:0]				SPACE[1:0]		PROG/DATA[1:0]		IDLE	IACK/CS
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CSORB — Chip-Select Option Register B **$YFFA66**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MODE	BYTE[1:0]		R/$\overline{W}$[1:0]		STRB	$\overline{DTACK}$[3:0]				SPACE[1:0]		PROG/DATA[1:0]		IDLE	UNUSED
RESET:															
ROMEN = 1															
0	0	0	1	1	0	1	1	1	0	1	1	0	0	0	0
ROMEN = 0															
0	1	1	1	1	0	1	1	1	0	1	1	0	0	0	0

CSORB, the option register for $\overline{CSB}$, contains reset values that support bootstrap operation from peripheral memory devices.

CSORC — Chip-Select Option Register C **$YFFA6A**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MODE	BYTE[1:0]		R/$\overline{W}$[1:0]		STRB	$\overline{DTACK}$[3:0]				SPACE[1:0]		PROG/DATA[1:0]		IDLE	UNUSED
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MODE — Asynchronous/Synchronous Mode

0 = Asynchronous mode selected

1 = Synchronous mode selected

In asynchronous mode, the chip-select is asserted synchronized with $\overline{AS}$ or $\overline{DS}$. In synchronous mode, the chip select is asserted synchronized with the optional ECLK pin. Although the MC68HC16V1 does not have ECLK, synchronous mode bus accesses can still be performed.

The $\overline{DTACK}$[3:0] field is not used in synchronous mode because a bus cycle is only performed as an asynchronous operation. When a match condition occurs on a chip-select programmed for synchronous operation, the chip select signals the EBI that an E-clock cycle is pending.

BYTE[1:0] — Upper/Lower Byte Option

This field is used to control the assertion of chip selects configured for 16-bit port accesses. Chip selects configured for 8-bit port accesses should use the BYTE = %11 (both bytes) encoding. BYTE = %00 disables the specified chip select. **Table 41** lists upper/lower byte options.

Table 41 Upper/Lower Byte Options

BYTE[1:0]	Description
00	Disable
01	Lower Byte
10	Upper Byte
11	Both Bytes

R/$\overline{W}$[1:0] — Read/Write

This field causes a chip select to be asserted only for read, only for write, or for both read and write cycles. R/$\overline{W}$ = %00 disables the specified chip-select. Refer to **Table 42**.

Table 42 R/$\overline{W}$ Encoding

R/$\overline{W}$[1:0]	Description
00	Disable
01	Read Only
10	Write Only
11	Read/Write

STRB — Address Strobe/Data Strobe

0 = Address strobe

1 = Data strobe

This bit controls the timing for assertion of a chip select in asynchronous mode. Selecting address strobe causes chip select to be asserted synchronized with $\overline{AS}$. Selecting data strobe causes chip select to be asserted synchronized with $\overline{DS}$.

$\overline{DTACK}$[3:0] — Data Transfer Acknowledge

This field specifies the source of $\overline{DTACK}$ in asynchronous mode. Bus timing can be adjusted with internal $\overline{DTACK}$ generation by controlling the number of wait states that are inserted in a bus cycle. **Table 43** shows the $\overline{DTACK}$ field encoding.

Table 43 $\overline{\text{DTACK}}$ Field Encoding

$\overline{\text{DTACK}}$[3:0]	Clock Cycles Required Per Access	Wait States Inserted Per Access
0000	2[1]	0
0001	3	1
0010	4	2
0011	5	3
0100	6	4
0101	7	5
0110	8	6
0111	9	7
1000	10	8
1001	11	9
1010	12	10
1011	13	11
1100	14	12
1101	15	13
1110	16	14
1111	—	External $\overline{\text{DTACK}}$

1. The SLIM can only perform a two-clock access when running in non-multiplexed mode. Two-clock accesses are not supported in multiplexed mode, and the SLIM may not properly execute such bus cycles if they are attempted.

SPACE[1:0] — Address Space

This option field allows the chip select to respond to address spaces which are indicated by the function codes generated by the CPU. The CPU16 normally operates in supervisor space, but interrupt acknowledge must take place in CPU space. **Table 44** shows SPACE field encoding.

Table 44 SPACE Field Encoding

SPACE[1:0]	Address Space
00	CPU Space
01	User Space[1]
10	Supervisor Space
11	Supervisor/User Space[1]

1. The CPU16 executes code only in supervisor mode, therefore, this space field encoding has no effect. Supervisor/user space is equivalent to supervisor space encoding.

PROG/DATA[1:0] — Program/Data Space

The PROG/DATA bits are defined to decode program and data space as shown in **Table 44**.

Table 45 Program/Data Field Encoding

PROG/DATA[1:0]	Program/Data Space
00	Data or Program
01	Data Space
10	Program Space
11	Reserved

This field only affects the response of chip selects and does not affect interrupt recognition by the CPU.

IDLE — Add Idle Clock

This field determines whether an idle clock cycle is added at the end of an external cycle to allow time for an external device to stop driving the bus before the next address is driven on the bus by the SLIM.

0 = No idle clock cycle added

1 = One idle clock cycle added

IACK/CS — Assert $\overline{\text{CSA}}$ on IACK

This bit, with IACK in CSCR, determines whether external IACK cycles are terminated externally or internally and whether or not $\overline{\text{CSA}}$ is also asserted during an external IACK cycle. **Table 46** shows the IACK/CS field.

Table 46 IACK/CS Field

IACK/CS	Description
0	$\overline{\text{CSA}}$ not asserted with external IACK cycle
1	$\overline{\text{CSA}}$ asserted with external IACK cycle if IACK/CS in CSCR = 1

3.10 Factory Test Block

The test submodule supports scan-based testing of the various MCU modules. It is integrated into the SLIM to support production test.

3.10.1 Test Registers

Test submodule registers are intended for Motorola use. Register names and addresses are provided to indicate that these addresses are occupied.

SLIMTR — Stream-Lined Integration Module Test Register **$YFFA02**

SLIMTRE — Stream-Lined Integration Module Test Register (E Clock) **$YFFA08**

TSTMSRA — Master Shift Register A **$YFFA40**

TSTMSRB — Master Shift Register B **$YFFA42**

TSTSC — Test Module Shift Count **$YFFA44**

TSTRC — Test Module Repetition Count **$YFFA46**

CREG — Test Submodule Control Register **$YFFA48**

DREG — Distributed Register **$YFFA4A**

4 Central Processing Unit

The CPU16 is a true 16-bit, high-speed device. It was designed to give M68HC11 users a path to higher performance while maintaining maximum compatibility with existing systems.

4.1 Overview

Ease of programming is an important consideration when using a microcontroller. The CPU16 instruction set is optimized for high performance. There are two 16-bit general-purpose accumulators and three 16-bit index registers. The CPU16 supports 8-bit (byte), 16-bit (word), and 32-bit (long-word) load and store operations, as well as 16- and 32-bit signed fractional operations. Code development is simplified by the background debugging mode.

CPU16 memory space includes a 1-Mbyte data space and a 1-Mbyte program space. Twenty-bit addressing and transparent bank switching are used to implement extended memory. In addition, most instructions automatically handle bank boundaries.

The CPU16 includes instructions and hardware to implement control-oriented digital signal processing functions with a minimum of interfacing. A multiply and accumulate unit provides the capability to multiply signed 16-bit fractional numbers and store the resulting 32-bit fixed point product in a 36-bit accumulator. Modulo addressing supports finite impulse response filters.

Use of high-level languages is increasing as controller applications become more complex and control programs become larger. These languages make rapid development of portable software possible. The CPU16 instruction set supports high-level languages.

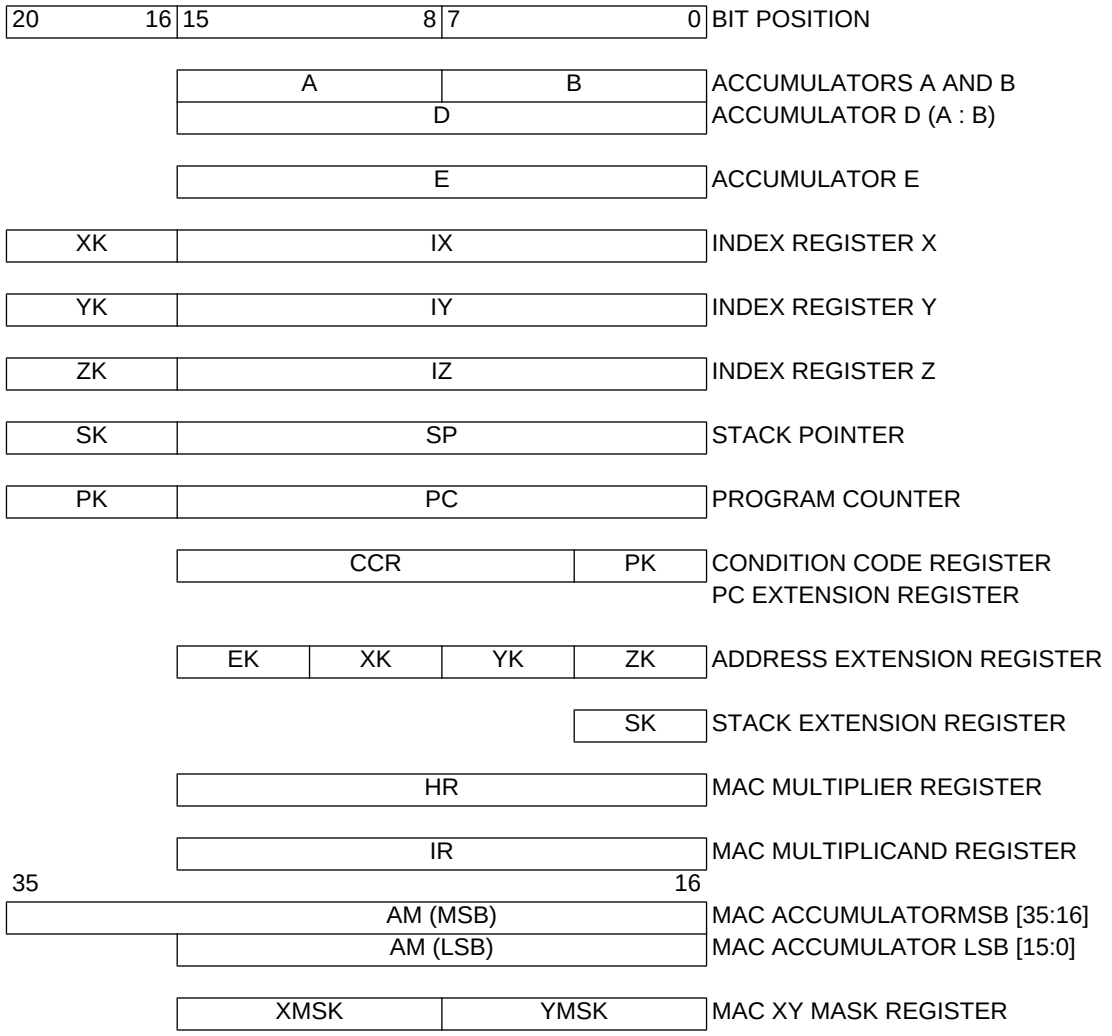
4.2 M68HC11 Compatibility

CPU16 architecture is a superset of M68HC11 CPU architecture. All M68HC11 CPU resources are available in the CPU16. M68HC11 CPU instructions are either directly implemented in the CPU16, or have been replaced by instructions with an equivalent form. The instruction sets are source code compatible, but some instructions are executed differently in the CPU16. These instructions are mainly related to interrupt and exception processing — M68HC11 CPU code that processes interrupts, handles stack frames, or manipulates the condition code register must be rewritten.

CPU16 execution times and number of cycles for all instructions are different from those of the M68HC11 CPU. As a result, cycle-related delays and timed control routines may be affected.

The CPU16 also has several new or enhanced addressing modes. M68HC11 CPU direct mode addressing has been replaced by a special form of indexed addressing that uses the new IZ register and a reset vector to provide greater flexibility.

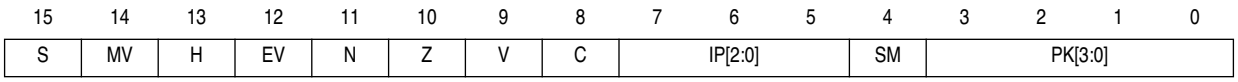
4.3 Programming Model



Accumulator A — 8-bit general-purpose register
Accumulator B — 8-bit general-purpose register
Accumulator D — 16-bit general-purpose register formed by concatenating accumulators A and B
Accumulator E — 16-bit general-purpose register
Index Register X — 16-bit indexing register, addressing extended by XK field in K register
Index Register Y — 16-bit indexing register, addressing extended by YK field in K register
Index Register Z — 16-bit indexing register, addressing extended by ZK field in K register
Stack Pointer — 16-bit dedicated register, addressing extended by the SK register
Program Counter — 16-bit dedicated register, addressing extended by PK field in CCR
Condition Code Register — 16-bit register containing condition flags, interrupt priority mask, and the program counter address extension field
K Register — 16-bit register made up of four 4-bit address extension fields
SK Register — 4-bit register containing the stack pointer address extension field
H Register — 16-bit multiply and accumulate input (multiplier) register
I Register — 16-bit multiply and accumulate input (multiplicand) register
MAC Accumulator — 36-bit multiply and accumulate result register
XMSK, YMSK — Determine which bits change when an offset is added

Figure 15 CPU16 Programming Model

4.3.1 Condition Code Register



S — STOP Enable
0 = Stop clock when LPSTOP instruction is executed
1 = Perform NOP when LPSTOP instruction is executed

MV — Accumulator M overflow flag
MV is set when an overflow into AM35 has occurred.

H — Half Carry Flag
H is set when a carry from A3 or B3 occurs during BCD addition.

EV — Extension Bit Overflow Flag
EV is set when an overflow into AM31 has occurred.

N — Negative Flag
N is set when the MSB of a result register is set.

Z — Zero Flag
Z is set when all bits of a result register are zero.

V — Overflow Flag
V is set when a two's complement overflow occurs as the result of an operation.

C — Carry Flag
C is set when a carry or borrow occurs during an arithmetic operation. This flag is also used during shift and rotate to facilitate multiple word operations.

IP[2:0] — Interrupt Priority Field
The priority value in this field (0 to 7) is used to mask interrupts.

SM — Saturate Mode Bit
When SM is set, if either EV or MV is set, data read from AM using TMER or TMET is given maximum positive or negative value, depending on the state of the AM sign bit before overflow.

PK[3:0] — Program Counter Address Extension Field
This field is concatenated with the program counter to form a 20-bit address.

4.4 Data Types

The CPU16 supports the following data types:

- Bit data
- 8-bit (byte) and 16-bit (word) integers
- 32-bit long integers
- 16-bit and 32-bit signed fractions (MAC operations only)
- 20-bit effective address consisting of 16-bit page address plus 4-bit extension

A byte is 8 bits wide and can be accessed at any byte location. A word is composed of two consecutive bytes, and is addressed at the lower byte. Instruction fetches are always accessed on word boundaries. Word operands are normally accessed on word boundaries as well, but can be accessed on odd byte boundaries, with a substantial performance penalty.

To be compatible with the M68HC11, misaligned word transfers and misaligned stack accesses are allowed. Transferring a misaligned word requires two successive byte operations.

4.5 Addressing Modes

The CPU16 provides ten types of addressing. Each type encompasses one or more addressing modes. Six CPU16 addressing types are identical to M68HC11 addressing types.

All modes generate ADDR[15:0]. This address is combined with ADDR[19:16] from an extension field to form a 20-bit effective address. Extension fields are part of a bank switching scheme that provides the CPU16 with a 1-Mbyte address space. Bank switching is transparent to most instructions. ADDR[19:16] of the effective address change when an access crosses a bank boundary. However, it is important to note that the value of the associated extension field is dependent on the type of instruction, and usually does not change as a result of effective address calculation.

In the immediate modes, the instruction argument is contained in bytes or words immediately following the instruction. The effective address is the address of the byte following the instruction. The AIS, AIX/Y/Z, ADDD and ADDE instructions have an extended 8-bit mode where the immediate value is an 8-bit signed number that is sign-extended to 16 bits, and then added to the appropriate register. Use of the extended 8-bit mode decreases execution time.

Extended mode instructions contain ADDR[15:0] in the word following the opcode. The effective address is formed by concatenating EK and the 16-bit extension.

In the indexed modes, registers IX, IY, and IZ, together with their associated extension fields, are used to calculate the effective address. Signed 16-bit mode and signed 20-bit mode are extensions to the M68HC11 indexed addressing mode.

For 8-bit indexed mode, an 8-bit unsigned offset contained in the instruction is added to the value contained in the index register and its associated extension field.

For 16-bit mode, a 16-bit signed offset contained in the instruction is added to the value contained in the index register and its associated extension field.

For 20-bit mode, a 20-bit signed offset is added to the value contained in the index register. This mode is used for JMP and JSR instructions.

Inherent mode instructions use information available to the processor to determine the effective address. Operands (if any) are system resources and are thus not fetched from memory.

Accumulator offset mode adds the contents of 16-bit accumulator E to one of the index registers and its associated extension field to form the effective address. This mode allows use of index registers and an accumulator within loops without corrupting accumulator D.

Relative modes are used for branch and long branch instructions. A byte or word signed two's complement offset is added to the program counter if the branch condition is satisfied. The new PC value, concatenated with the PK field, is the effective address.

Post-modified index mode is used with the MOVB and MOVW instructions. A signed 8-bit offset is added to index register X after the effective address formed by XK and IX is used.

In M68HC11 systems, direct mode can be used to perform rapid accesses to RAM or I/O mapped into page 0 ($0000 to $00FF), but the CPU16 uses the first 512 bytes of page 0 for exception vectors. To compensate for the loss of direct mode, the ZK field and index register Z have been assigned reset initialization vectors. By resetting the ZK field to a chosen page, and using 8-bit unsigned index mode with IZ, a programmer can access useful data structures anywhere in the address map.

4.6 Instruction Set

The CPU16 instruction set is based on that of the M68HC11, but the opcode map has been rearranged to maximize performance with a 16-bit data bus. All M68HC11 instructions are supported by the CPU16, although they may be executed differently. Most M68HC11 code runs on the CPU16 following reassembly. However, take into account changed instruction times, the interrupt mask, and the new interrupt stack frame.

The CPU16 has a full range of 16-bit arithmetic and logic instructions, including signed and unsigned multiplication and division. New instructions have been added to support extended addressing and digital signal processing.

Table 47 is a quick reference to the entire CPU16 instruction set. Because it is only affected by a few instructions, the LSB of the condition code register is not shown in the table. Instructions that affect the interrupt mask and PK field are noted. **Table 48** provides a key to the table nomenclature.

Table 47 Instruction Set Summary

Mnemonic	Operation	Description	Address	Instruction			Condition Codes							
			Mode	Opcode	Operand	Cycles	S	MV	H	EV	N	Z	V	C
ABA	Add B to A	(A) + (B) ⇒ A	INH	370B	—	2	—	—	Δ	—	Δ	Δ	Δ	Δ
ABX	Add B to IX	(XK : IX) + (000 : B) ⇒ XK : IX	INH	374F	—	2	—	—	—	—	—	—	—	—
ABY	Add B to IY	(YK : IY) + (000 : B) ⇒ YK : IY	INH	375F	—	2	—	—	—	—	—	—	—	—
ABZ	Add B to IZ	(ZK : IZ) + (000 : B) ⇒ ZK : IZ	INH	376F	—	2	—	—	—	—	—	—	—	—
ACE	Add E to AM	(AM[31:16]) + (E) ⇒ AM	INH	3722	—	2	—	Δ	—	Δ	—	—	—	—
ACED	Add E : D to AM	(AM) + (E : D) ⇒ AM	INH	3723	—	4	—	Δ	—	Δ	—	—	—	—
ADCA	Add with Carry to A	(A) + (M) + C ⇒ A	IND8, X	43	ff	6	—	—	Δ	—	Δ	Δ	Δ	Δ
			IND8, Y	53	ff	6								
			IND8, Z	63	ff	6								
			IMM8	73	ii	2								
			IND16, X	1743	gggg	6								
			IND16, Y	1753	gggg	6								
			IND16, Z	1763	gggg	6								
			EXT	1773	hh ll	6								
			E, X	2743	—	6								
			E, Y	2753	—	6								
			E, Z	2763	—	6								
ADCB	Add with Carry to B	(B) + (M) + C ⇒ B	IND8, X	C3	ff	6	—	—	Δ	—	Δ	Δ	Δ	Δ
			IND8, Y	D3	ff	6								
			IND8, Z	E3	ff	6								
			IMM8	F3	ii	2								
			IND16, X	17C3	gggg	6								
			IND16, Y	17D3	gggg	6								
			IND16, Z	17E3	gggg	6								
			EXT	17F3	hh ll	6								
			E, X	27C3	—	6								
			E, Y	27D3	—	6								
			E, Z	27E3	—	6								
ADCD	Add with Carry to D	(D) + (M : M + 1) + C ⇒ D	IND8, X	83	ff	6	—	—	—	—	Δ	Δ	Δ	Δ
			IND8, Y	93	ff	6								
			IND8, Z	A3	ff	6								
			IMM16	37B3	jj kk	4								
			IND16, X	37C3	gggg	6								
			IND16, Y	37D3	gggg	6								
			IND16, Z	37E3	gggg	6								
			EXT	37F3	hh ll	6								
			E, X	2783	—	6								
			E, Y	2793	—	6								
			E, Z	27A3	—	6								
ADCE	Add with Carry to E	(E) + (M : M + 1) + C ⇒ E	IMM16	3733	jj kk	4	—	—	—	—	Δ	Δ	Δ	Δ
			IND16, X	3743	gggg	6								
			IND16, Y	3753	gggg	6								
			IND16, Z	3763	gggg	6								
			EXT	3773	hh ll	6								
ADDA	Add to A	(A) + (M) ⇒ A	IND8, X	41	ff	6	—	—	Δ	—	Δ	Δ	Δ	Δ
			IND8, Y	51	ff	6								
			IND8, Z	61	ff	6								
			IMM8	71	ii	2								
			IND16, X	1741	gggg	6								
			IND16, Y	1751	gggg	6								
			IND16, Z	1761	gggg	6								
			EXT	1771	hh ll	6								
			E, X	2741	—	6								
			E, Y	2751	—	6								
			E, Z	2761	—	6								

Table 47 Instruction Set Summary (Continued)

Mnemonic	Operation	Description	Address Mode	Instruction Opcode	Instruction Operand	Instruction Cycles	Condition Codes S	MV	H	EV	N	Z	V	C
ADDB	Add to B	(B) + (M) ⇒ B	IND8, X	C1	ff	6	—	—	Δ	—	Δ	Δ	Δ	Δ
			IND8, Y	D1	ff	6								
			IND8, Z	E1	ff	6								
			IMM8	F1	ii	2								
			IND16, X	17C1	gggg	6								
			IND16, Y	17D1	gggg	6								
			IND16, Z	17E1	gggg	6								
			EXT	17F1	hh ll	6								
			E, X	27C1	—	6								
			E, Y	27D1	—	6								
			E, Z	27E1	—	6								
ADDD	Add to D	(D) + (M : M + 1) ⇒ D	IND8, X	81	ff	6	—	—	—	—	Δ	Δ	Δ	Δ
			IND8, Y	91	ff	6								
			IND8, Z	A1	ff	6								
			IMM8	FC	ii	2								
			IMM16	37B1	jj kk	4								
			IND16, X	37C1	gggg	6								
			IND16, Y	37D1	gggg	6								
			IND16, Z	37E1	gggg	6								
			EXT	37F1	hh ll	6								
			E, X	2781	—	6								
			E, Y	2791	—	6								
			E, Z	27A1	—	6								
ADDE	Add to E	(E) + (M : M + 1) ⇒ E	IMM8	7C	ii	2	—	—	—	—	Δ	Δ	Δ	Δ
			IMM16	3731	jj kk	4								
			IND16, X	3741	gggg	6								
			IND16, Y	3751	gggg	6								
			IND16, Z	3761	gggg	6								
			EXT	3771	hh ll	6								
ADE	Add D to E	(E) + (D) ⇒ E	INH	2778	—	2	—	—	—	—	Δ	Δ	Δ	Δ
ADX	Add D to IX	(XK : IX) + (20 « D) ⇒ XK : IX	INH	37CD	—	2	—	—	—	—	—	—	—	—
ADY	Add D to IY	(YK : IY) + (20 « D) ⇒ YK : IY	INH	37DD	—	2	—	—	—	—	—	—	—	—
ADZ	Add D to IZ	(ZK : IZ) + (20 « D) ⇒ ZK : IZ	INH	37ED	—	2	—	—	—	—	—	—	—	—
AEX	Add E to IX	(XK : IX) + (20 « D) ⇒ XK : IX	INH	374D	—	2	—	—	—	—	—	—	—	—
AEY	Add E to IY	(YK : IY) + (20 « D) ⇒ YK : IY	INH	375D	—	2	—	—	—	—	—	—	—	—
AEZ	Add E to IZ	(ZK : IZ) + (20 « D) ⇒ ZK : IZ	INH	376D	—	2	—	—	—	—	—	—	—	—
AIS	Add Immediate Data to Stack Pointer	(SK : SP) + (20 « IMM) ⇒ SK : SP	IMM8	3F	ii	2	—	—	—	—	—	—	—	—
			IMM16	373F	jj kk	4								
AIX	Add Immediate Value to IX	(XK : IX) + (20 « IMM) ⇒ XK : IX	IMM8	3C	ii	2	—	—	—	—	—	Δ	—	—
			IMM16	373C	jj kk	4								
AIY	Add Immediate Value to IY	(YK : IY) + (20 « IMM) ⇒ YK : IY	IMM8	3D	ii	2	—	—	—	—	—	Δ	—	—
			IMM16	373D	jj kk	4								
AIZ	Add Immediate Value to IZ	(ZK : IZ) + (20 « IMM) ⇒ ZK : IZ	IMM8	3E	ii	2	—	—	—	—	—	Δ	—	—
			IMM16	373E	jj kk	4								
ANDA	AND A	(A) • (M) ⇒ A	IND8, X	46	ff	6	—	—	—	—	Δ	Δ	0	—
			IND8, Y	56	ff	6								
			IND8, Z	66	ff	6								
			IMM8	76	ii	2								
			IND16, X	1746	gggg	6								
			IND16, Y	1756	gggg	6								
			IND16, Z	1766	gggg	6								
			EXT	1776	hh ll	6								
			E, X	2746	—	6								
			E, Y	2756	—	6								
			E, Z	2766	—	6								

Table 47 Instruction Set Summary (Continued)

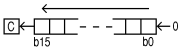
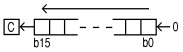
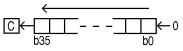
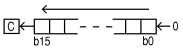
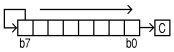
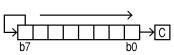
Mnemonic	Operation	Description	Address Mode	Instruction Opcode	Instruction Operand	Instruction Cycles	S	MV	H	EV	N	Z	V	C
ANDB	AND B	(B) • (M) ⇒ B	IND8, X	C6	ff	6	—	—	—	—	Δ	Δ	0	—
			IND8, Y	D6	ff	6								
			IND8, Z	E6	ff	6								
			IMM8	F6	ii	2								
			IND16, X	17C6	gggg	6								
			IND16, Y	17D6	gggg	6								
			IND16, Z	17E6	gggg	6								
			EXT	17F6	hh ll	6								
			E, X	27C6	—	6								
			E, Y	27D6	—	6								
			E, Z	27E6	—	6								
ANDD	AND D	(D) • (M : M + 1) ⇒ D	IND8, X	86	ff	6	—	—	—	—	Δ	Δ	0	—
			IND8, Y	96	ff	6								
			IND8, Z	A6	ff	6								
			IMM16	37B6	jj kk	4								
			IND16, X	37C6	gggg	6								
			IND16, Y	37D6	gggg	6								
			IND16, Z	37E6	gggg	6								
			EXT	37F6	hh ll	6								
			E, X	2786	—	6								
			E, Y	2796	—	6								
			E, Z	27A6	—	6								
ANDE	AND E	(E) • (M : M + 1) ⇒ E	IMM16	3736	jj kk	4	—	—	—	—	Δ	Δ	0	—
			IND16, X	3746	gggg	6								
			IND16, Y	3756	gggg	6								
			IND16, Z	3766	gggg	6								
			EXT	3776	hh ll	6								
ANDP[1]	AND CCR	(CCR) • IMM16⇒ CCR	IMM16	373A	jj kk	4	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ
ASL	Arithmetic Shift Left		IND8, X	04	ff	8	—	—	—	—	Δ	Δ	Δ	Δ
			IND8, Y	14	ff	8								
			IND8, Z	24	ff	8								
			IND16, X	1704	gggg	8								
			IND16, Y	1714	gggg	8								
			IND16, Z	1724	gggg	8								
			EXT	1734	hh ll	8								
ASLA	Arithmetic Shift Left A		INH	3704	—	2	—	—	—	—	Δ	Δ	Δ	Δ
ASLB	Arithmetic Shift Left B		INH	3714	—	2	—	—	—	—	Δ	Δ	Δ	Δ
ASLD	Arithmetic Shift Left D		INH	27F4	—	2	—	—	—	—	Δ	Δ	Δ	Δ
ASLE	Arithmetic Shift Left E		INH	2774	—	2	—	—	—	—	Δ	Δ	Δ	Δ
ASLM	Arithmetic Shift Left AM		INH	27B6	—	4	—	Δ	—	Δ	Δ	—	—	Δ
ASLW	Arithmetic Shift Left Word		IND16, X	2704	gggg	8	—	—	—	—	Δ	Δ	Δ	Δ
			IND16, Y	2714	gggg	8								
			IND16, Z	2724	gggg	8								
			EXT	2734	hh ll	8								
ASR	Arithmetic Shift Right		IND8, X	0D	ff	8	—	—	—	—	Δ	Δ	Δ	Δ
			IND8, Y	1D	ff	8								
			IND8, Z	2D	ff	8								
			IND16, X	170D	gggg	8								
			IND16, Y	171D	gggg	8								
			IND16, Z	172D	gggg	8								
			EXT	173D	hh ll	8								
ASRA	Arithmetic Shift Right A		INH	370D	—	2	—	—	—	—	Δ	Δ	Δ	Δ

Table 47 Instruction Set Summary (Continued)

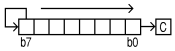
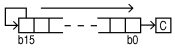
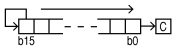
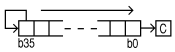
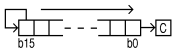
Mnemonic	Operation	Description	Address Mode	Instruction Opcode	Instruction Operand	Instruction Cycles	Condition Codes S	MV	H	EV	N	Z	V	C
ASRB	Arithmetic Shift Right B		INH	371D	—	2	—	—	—	—	Δ	Δ	Δ	Δ
ASRD	Arithmetic Shift Right D		INH	27FD	—	2	—	—	—	—	Δ	Δ	Δ	Δ
ASRE	Arithmetic Shift Right E		INH	277D	—	2	—	—	—	—	Δ	Δ	Δ	Δ
ASRM	Arithmetic Shift Right AM		INH	27BA	—	4	—	—	—	Δ	Δ	—	—	Δ
ASRW	Arithmetic Shift Right Word		IND16, X IND16, Y IND16, Z EXT	270D 271D 272D 273D	gggg gggg gggg hh ll	8 8 8 8	—	—	—	—	Δ	Δ	Δ	Δ
BCC[2]	Branch if Carry Clear	If C = 0, branch	REL8	B4	rr	6, 2	—	—	—	—	—	—	—	—
BCLR	Clear Bit(s)	(M) • (Mask) ⇒ M	IND8, X IND8, Y IND8, Z IND16, X IND16, Y IND16, Z EXT	1708 1718 1728 08 18 28 38	mm ff mm ff mm ff mm gggg mm gggg mm gggg mm hh ll	8 8 8 8 8 8 8	—	—	—	—	Δ	Δ	0	—
BCLRW	Clear Bit(s) in a Word	(M : M + 1) • (Mask) ⇒ M : M + 1	IND16, X IND16, Y IND16, Z EXT	2708 2718 2728 2738	gggg mmmm gggg mmmm gggg mmmm hh ll mmmm	10 10 10 10	—	—	—	—	Δ	Δ	0	—
BCS[2]	Branch if Carry Set	If C = 1, branch	REL8	B5	rr	6, 2	—	—	—	—	—	—	—	—
BEQ[2]	Branch if Equal	If Z = 1, branch	REL8	B7	rr	6, 2	—	—	—	—	—	—	—	—
BGE[2]	Branch if Greater Than or Equal to Zero	If N ⊕ V = 0, branch	REL8	BC	rr	6, 2	—	—	—	—	—	—	—	—
BGND	Enter Background Debug Mode	If BDM enabled, begin debug; else, illegal instruction trap	INH	37A6	—	—	—	—	—	—	—	—	—	—
BGT[2]	Branch if Greater Than Zero	If Z ✣ (N ⊕ V) = 0, branch	REL8	BE	rr	6, 2	—	—	—	—	—	—	—	—
BHI[2]	Branch if Higher	If C ✣ Z = 0, branch	REL8	B2	rr	6, 2	—	—	—	—	—	—	—	—
BITA	Bit Test A	(A) • (M)	IND8, X IND8, Y IND8, Z IMM8 IND16, X IND16, Y IND16, Z EXT E, X E, Y E, Z	49 59 69 79 1749 1759 1769 1779 2749 2759 2769	ff ff ff ii gggg gggg gggg hh ll — — —	6 6 6 2 6 6 6 6 6 6 6	—	—	—	—	Δ	Δ	0	—
BITB	Bit Test B	(B) • (M)	IND8, X IND8, Y IND8, Z IMM8 IND16, X IND16, Y IND16, Z EXT E, X E, Y E, Z	C9 D9 E9 F9 17C9 17D9 17E9 17F9 27C9 27D9 27E9	ff ff ff ii gggg gggg gggg hh ll — — —	6 6 6 2 6 6 6 6 6 6 6	—	—	—	—	Δ	Δ	0	—

Table 47 Instruction Set Summary (Continued)

Mnemonic	Operation	Description	Address Mode	Instruction Opcode	Instruction Operand	Instruction Cycles	Condition Codes S	MV	H	EV	N	Z	V	C
BLE[2]	Branch if Less Than or Equal to Zero	If Z ✣ (N ⊕ V) = 1, branch	REL8	BF	rr	6, 2	—	—	—	—	—	—	—	—
BLS[2]	Branch if Lower or Same	If C ✣ Z = 1, branch	REL8	B3	rr	6, 2	—	—	—	—	—	—	—	—
BLT[2]	Branch if Less Than Zero	If N ⊕ V = 1, branch	REL8	BD	rr	6, 2	—	—	—	—	—	—	—	—
BMI[2]	Branch if Minus	If N = 1, branch	REL8	BB	rr	6, 2	—	—	—	—	—	—	—	—
BNE[2]	Branch if Not Equal	If Z = 0, branch	REL8	B6	rr	6, 2	—	—	—	—	—	—	—	—
BPL[2]	Branch if Plus	If N = 0, branch	REL8	BA	rr	6, 2	—	—	—	—	—	—	—	—
BRA	Branch Always	If 1 = 1, branch	REL8	B0	rr	6	—	—	—	—	—	—	—	—
BRCLR[2]	Branch if Bit(s) Clear	If (M) • (Mask) = 0, branch	IND8, X IND8, Y IND8, Z IND16, X IND16, Y IND16, Z EXT	CB DB EB 0A 1A 2A 3A	mm ff rr mm ff rr mm ff rr mm gggg rrrr mm gggg rrrr mm gggg rrrr mm hh ll rrrr	10, 12 10, 12 10, 12 10, 14 10, 14 10, 14 10, 14	—	—	—	—	—	—	—	—
BRN	Branch Never	If 1 = 0, branch	REL8	B1	rr	2	—	—	—	—	—	—	—	—
BRSET[2]	Branch if Bit(s) Set	If (M) • (Mask) = 0, branch	IND8, X IND8, Y IND8, Z IND16, X IND16, Y IND16, Z EXT	8B 9B AB 0B 1B 2B 3B	mm ff rr mm ff rr mm ff rr mm gggg rrrr mm gggg rrrr mm gggg rrrr mm hh ll rrrr	10, 12 10, 12 10, 12 10, 14 10, 14 10, 14 10, 14	—	—	—	—	—	—	—	—
BSET	Set Bit(s)	(M) ✣ (Mask) ⇒ M	IND8, X IND8, Y IND8, Z IND16, X IND16, Y IND16, Z EXT	1709 1719 1729 09 19 29 39	mm ff mm ff mm ff mm gggg mm gggg mm gggg mm hh ll	8 8 8 8 8 8 8	—	—	—	—	Δ	Δ	0	Δ
BSETW	Set Bit(s) in Word	(M : M + 1) ✣ (Mask) ⇒ M : M + 1	IND16, X IND16, Y IND16, Z EXT	2709 2719 2729 2739	gggg mmmm gggg mmmm gggg mmmm hh ll mmmm	10 10 10 10	—	—	—	—	Δ	Δ	0	Δ
BSR	Branch to Subroutine	(PK : PC) - 2 ⇒ PK : PC Push (PC) (SK : SP) - 2 ⇒ SK : SP Push (CCR) (SK : SP) - 2 ⇒ SK : SP (PK : PC) + Offset ⇒ PK : PC	REL8	36	rr	10	—	—	—	—	—	—	—	—
BVC[2]	Branch if Overflow Clear	If V = 0, branch	REL8	B8	rr	6, 2	—	—	—	—	—	—	—	—
BVS[2]	Branch if Overflow Set	If V = 1, branch	REL8	B9	rr	6, 2	—	—	—	—	—	—	—	—
CBA	Compare A to B	(A) – (B)	INH	371B	—	2	—	—	—	—	Δ	Δ	Δ	Δ
CLR	Clear a Byte in Memory	$00 ⇒ M	IND8, X IND8, Y IND8, Z IND16, X IND16, Y IND16, Z EXT	05 15 25 1705 1715 1725 1735	ff ff ff gggg gggg gggg hh ll	4 4 4 6 6 6 6	—	—	—	—	0	1	0	0

Table 47 Instruction Set Summary (Continued)

Mnemonic	Operation	Description	Address Mode	Instruction Opcode	Instruction Operand	Instruction Cycles	Condition Codes S	MV	H	EV	N	Z	V	C
CLRA	Clear A	$00 ⇒ A	INH	3705	—	2	—	—	—	—	0	1	0	0
CLRB	Clear B	$00 ⇒ B	INH	3715	—	2	—	—	—	—	0	1	0	0
CLRD	Clear D	$0000 ⇒ D	INH	27F5	—	2	—	—	—	—	0	1	0	0
CLRE	Clear E	$0000 ⇒ E	INH	2775	—	2	—	—	—	—	0	1	0	0
CLRM	Clear AM	$000000000 ⇒ AM[35:0]	INH	27B7	—	2	—	0	—	0	—	—	—	—
CLRW	Clear a Word in Memory	$0000 ⇒ M : M + 1	IND16, X IND16, Y IND16, Z EXT	2705 2715 2725 2735	gggg gggg gggg hh ll	6 6 6 6	—	—	—	—	0	1	0	0
CMPA	Compare A to Memory	(A) – (M)	IND8, X IND8, Y IND8, Z IMM8 IND16, X IND16, Y IND16, Z EXT E, X E, Y E, Z	48 58 68 78 1748 1758 1768 1778 2748 2758 2768	ff ff ff ii gggg gggg gggg hh ll — — —	6 6 6 2 6 6 6 6 6 6 6	—	—	—	—	Δ	Δ	Δ	Δ
CMPB	Compare B to Memory	(B) – (M)	IND8, X IND8, Y IND8, Z IMM8 IND16, X IND16, Y IND16, Z EXT E, X E, Y E, Z	C8 D8 E8 F8 17C8 17D8 17E8 17F8 27C8 27D8 27E8	ff ff ff ii gggg gggg gggg hh ll — — —	6 6 6 2 6 6 6 6 6 6 6	—	—	—	—	Δ	Δ	Δ	Δ
COM	One's Complement	$FF – (M) ⇒ M, or $\overline{M}$ ⇒ M	IND8, X IND8, Y IND8, Z IND16, X IND16, Y IND16, Z EXT	00 10 20 1700 1710 1720 1730	ff ff ff gggg gggg gggg hh ll	8 8 8 8 8 8 8	—	—	—	—	Δ	Δ	0	1
COMA	One's Complement A	$FF – (A) ⇒ A, or $\overline{M}$ ⇒ A	INH	3700	—	2	—	—	—	—	Δ	Δ	0	1
COMB	One's Complement B	$FF – (B) ⇒ B, or $\overline{B}$ ⇒ B	INH	3710	—	2	—	—	—	—	Δ	Δ	0	1
COMD	One's Complement D	$FFFF – (D) ⇒ D, or $\overline{D}$ ⇒ D	INH	27F0	—	2	—	—	—	—	Δ	Δ	0	1
COME	One's Complement E	$FFFF – (E) ⇒ E, or $\overline{E}$ ⇒ E	INH	2770	—	2	—	—	—	—	Δ	Δ	0	1
COMW	One's Complement Word	$FFFF – M : M + 1 ⇒ M : M + 1, or ($\overline{M : M + 1}$) ⇒ M : M + 1	IND16, X IND16, Y IND16, Z EXT	2700 2710 2720 2730	gggg gggg gggg hh ll	8 8 8 8	—	—	—	—	Δ	Δ	0	1
CPD	Compare D to Memory	(D) – (M : M + 1)	IND8, X IND8, Y IND8, Z IMM16 IND16, X IND16, Y IND16, Z EXT E, X E, Y E, Z	88 98 A8 37B8 37C8 37D8 37E8 37F8 2788 2798 27A8	ff ff ff jj kk gggg gggg gggg hh ll — — —	6 6 6 4 6 6 6 6 6 6 6	—	—	—	—	Δ	Δ	Δ	Δ
CPE	Compare E to Memory	(E) – (M : M + 1)	IMM16 IND16, X IND16, Y IND16, Z EXT	3738 3748 3758 3768 3778	jjkk gggg gggg gggg hhll	4 6 6 6 6	—	—	—	—	Δ	Δ	Δ	Δ

Table 47 Instruction Set Summary (Continued)

Mnemonic	Operation	Description	Address Mode	Instruction Opcode	Instruction Operand	Instruction Cycles	Condition Codes S	MV	H	EV	N	Z	V	C
CPS	Compare Stack Pointer to Memory	(SP) – (M : M + 1)	IND8, X	4F	ff	6	—	—	—	—	Δ	Δ	Δ	Δ
			IND8, Y	5F	ff	6								
			IND8, Z	6F	ff	6								
			IMM16	377F	jj kk	4								
			IND16, X	174F	gggg	6								
			IND16, Y	175F	gggg	6								
			IND16, Z	176F	gggg	6								
			EXT	177F	hh ll	6								
CPX	Compare IX to Memory	(IX) – (M : M + 1)	IND8, X	4C	ff	6	—	—	—	—	Δ	Δ	Δ	Δ
			IND8, Y	5C	ff	6								
			IND8, Z	6C	ff	6								
			IMM16	377C	jj kk	4								
			IND16, X	174C	gggg	6								
			IND16, Y	175C	gggg	6								
			IND16, Z	176C	gggg	6								
			EXT	177C	hh ll	6								
CPY	Compare IY to Memory	(IY) – (M : M + 1)	IND8, X	4D	ff	6	—	—	—	—	Δ	Δ	Δ	Δ
			IND8, Y	5D	ff	6								
			IND8, Z	6D	ff	6								
			IMM16	377D	jj kk	4								
			IND16, X	174D	gggg	6								
			IND16, Y	175D	gggg	6								
			IND16, Z	176D	gggg	6								
			EXT	177D	hh ll	6								
CPZ	Compare IZ to Memory	(IZ) – (M : M + 1)	IND8, X	4E	ff	6	—	—	—	—	Δ	Δ	Δ	Δ
			IND8, Y	5E	ff	6								
			IND8, Z	6E	ff	6								
			IMM16	377E	jj kk	4								
			IND16, X	174E	gggg	6								
			IND16, Y	175E	gggg	6								
			IND16, Z	176E	gggg	6								
			EXT	177E	hh ll	6								
DAA	Decimal Adjust A	$(A)_{10}$	INH	3721	—	2	—	—	—	—	Δ	Δ	U	Δ
DEC	Decrement Memory	(M) – $01 ⇒ M	IND8, X	01	ff	8	—	—	—	—	Δ	Δ	Δ	—
			IND8, Y	11	ff	8								
			IND8, Z	21	ff	8								
			IND16, X	1701	gggg	8								
			IND16, Y	1711	gggg	8								
			IND16, Z	1721	gggg	8								
			EXT	1731	hh ll	8								
DECA	Decrement A	(A) – $01 ⇒ A	INH	3701	—	2	—	—	—	—	Δ	Δ	Δ	—
DECB	Decrement B	(B) – $01 ⇒ B	INH	3711	—	2	—	—	—	—	Δ	Δ	Δ	—
DECW	Decrement Memory Word	(M : M + 1) – $0001 ⇒ M : M + 1	IND16, X	2701	gggg	8	—	—	—	—	Δ	Δ	Δ	—
			IND16, Y	2711	gggg	8								
			IND16, Z	2721	gggg	8								
			EXT	2731	hh ll	8								
EDIV	Extended Unsigned Integer Divide	(E : D) / (IX) Quotient ⇒ IX Remainder ⇒ D	INH	3728	—	24	—	—	—	—	Δ	Δ	Δ	Δ
EDIVS	Extended Signed Integer Divide	(E : D) / (IX) Quotient ⇒ IX Remainder ⇒ D	INH	3729	—	38	—	—	—	—	Δ	Δ	Δ	Δ
EMUL	Extended Unsigned Multiply	(E) ∗ (D) ⇒ E : D	INH	3725	—	10	—	—	—	—	Δ	Δ	—	Δ
EMULS	Extended Signed Multiply	(E) ∗ (D) ⇒ E : D	INH	3726	—	8	—	—	—	—	Δ	Δ	—	Δ
EORA	Exclusive OR A	(A) ⊕ (M) ⇒ A	IND8, X	44	ff	6	—	—	—	—	Δ	Δ	0	—
			IND8, Y	54	ff	6								
			IND8, Z	64	ff	6								
			IMM8	74	ii	2								
			IND16, X	1744	gggg	6								
			IND16, Y	1754	gggg	6								
			IND16, Z	1764	gggg	6								
			EXT	1774	hh ll	6								
			E, X	2744	—	6								
			E, Y	2754	—	6								
			E, Z	2764	—	6								

Table 47 Instruction Set Summary (Continued)

Mnemonic	Operation	Description	Address Mode	Instruction Opcode	Instruction Operand	Instruction Cycles	S	MV	H	EV	N	Z	V	C
EORB	Exclusive OR B	(B) ⊕ (M) ⇒ B	IND8, X	C4	ff	6	—	—	—	—	Δ	Δ	0	—
			IND8, Y	D4	ff	6								
			IND8, Z	E4	ff	6								
			IMM8	F4	ii	2								
			IND16, X	17C4	gggg	6								
			IND16, Y	17D4	gggg	6								
			IND16, Z	17E4	gggg	6								
			EXT	17F4	hh ll	6								
			E, X	27C4	—	6								
			E, Y	27D4	—	6								
			E, Z	27E4	—	6								
EORD	Exclusive OR D	(D) ⊕ (M : M + 1) ⇒ D	IND8, X	84	ff	6	—	—	—	—	Δ	Δ	0	—
			IND8, Y	94	ff	6								
			IND8, Z	A4	ff	6								
			IMM16	37B4	jj kk	4								
			IND16, X	37C4	gggg	6								
			IND16, Y	37D4	gggg	6								
			IND16, Z	37E4	gggg	6								
			EXT	37F4	hh ll	6								
			E, X	2784	—	6								
			E, Y	2794	—	6								
			E, Z	27A4	—	6								
EORE	Exclusive OR E	(E) ⊕ (M : M + 1) ⇒ E	IMM16	3734	jj kk	4	—	—	—	—	Δ	Δ	0	—
			IND16, X	3744	gggg	6								
			IND16, Y	3754	gggg	6								
			IND16, Z	3764	gggg	6								
			EXT	3774	hh ll	6								
FDIV	Fractional Unsigned Divide	(D) / (IX) ⇒ IX Remainder ⇒ D	INH	372B	—	22	—	—	—	—	—	Δ	Δ	Δ
FMULS	Fractional Signed Multiply	(E) * (D) ⇒ E : D[31:1] 0 ⇒ D[0]	INH	3727	—	8	—	—	—	—	Δ	Δ	Δ	Δ
IDIV	Integer Divide	(D) / (IX) ⇒ IX Remainder ⇒ D	INH	372A	—	22	—	—	—	—	—	Δ	0	Δ
INC	Increment Memory	(M) + $01 ⇒ M	IND8, X	03	ff	8	—	—	—	—	Δ	Δ	Δ	—
			IND8, Y	13	ff	8								
			IND8, Z	23	ff	8								
			IND16, X	1703	gggg	8								
			IND16, Y	1713	gggg	8								
			IND16, Z	1723	gggg	8								
			EXT	1733	hh ll	8								
INCA	Increment A	(A) + $01 ⇒ A	INH	3703	—	2	—	—	—	—	Δ	Δ	Δ	—
INCB	Increment B	(B) + $01 ⇒ B	INH	3713	—	2	—	—	—	—	Δ	Δ	Δ	—
INCW	Increment Memory Word	(M : M + 1) + $0001 ⇒ M : M + 1	IND16, X	2703	gggg	8	—	—	—	—	Δ	Δ	Δ	—
			IND16, Y	2713	gggg	8								
			IND16, Z	2723	gggg	8								
			EXT	2733	hh ll	8								
JMP	Jump	⟨ea⟩ ⇒ PK : PC	EXT20	7A	zb hh ll	6	—	—	—	—	—	—	—	—
			IND20, X	4B	zg gggg	8								
			IND20, Y	5B	zg gggg	8								
			IND20, Z	6B	zg gggg	8								
JSR	Jump to Subroutine	Push (PC) (SK : SP) – $0002 ⇒ SK : SP Push (CCR) (SK : SP) – $0002 ⇒ SK : SP ⟨ea⟩ ⇒ PK : PC	EXT20	FA	zb hh ll	10	—	—	—	—	—	—	—	—
			IND20, X	89	zg gggg	12								
			IND20, Y	99	zg gggg	12								
			IND20, Z	A9	zg gggg	12								
LBCC[2]	Long Branch if Carry Clear	If C = 0, branch	REL16	3784	rrrr	6, 4	—	—	—	—	—	—	—	—
LBCS[2]	Long Branch if Carry Set	If C = 1, branch	REL16	3785	rrrr	6, 4	—	—	—	—	—	—	—	—
LBEQ[2]	Long Branch if Equal to Zero	If Z = 1, branch	REL16	3787	rrrr	6, 4	—	—	—	—	—	—	—	—
LBEV[2]	Long Branch if EV Set	If EV = 1, branch	REL16	3791	rrrr	6, 4	—	—	—	—	—	—	—	—
LBGE[2]	Long Branch if Greater Than or Equal to Zero	If N ⊕ V = 0, branch	REL16	378C	rrrr	6, 4	—	—	—	—	—	—	—	—
LBGT[2]	Long Branch if Greater Than Zero	If Z + (N ⊕ V) = 0, branch	REL16	378E	rrrr	6, 4	—	—	—	—	—	—	—	—

Table 47 Instruction Set Summary (Continued)

Mnemonic	Operation	Description	Address Mode	Instruction Opcode	Instruction Operand	Instruction Cycles	Condition Codes S	MV	H	EV	N	Z	V	C
LBHI [2]	Long Branch if Higher	If C ✣ Z = 0, branch	REL16	3782	rrrr	6, 4	—	—	—	—	—	—	—	—
LBLE[2]	Long Branch if Less Than or Equal to Zero	If Z ✣ (N ⊕ V) = 1, branch	REL16	378F	rrrr	6, 4	—	—	—	—	—	—	—	—
LBLS[2]	Long Branch if Lower or Same	If C ✣ Z = 1, branch	REL16	3783	rrrr	6, 4	—	—	—	—	—	—	—	—
LBLT[2]	Long Branch if Less Than Zero	If N ⊕ V = 1, branch	REL16	378D	rrrr	6, 4	—	—	—	—	—	—	—	—
LBMI[2]	Long Branch if Minus	If N = 1, branch	REL16	378B	rrrr	6, 4	—	—	—	—	—	—	—	—
LBMV[2]	Long Branch if MV Set	If MV = 1, branch	REL16	3790	rrrr	6, 4	—	—	—	—	—	—	—	—
LBNE[2]	Long Branch if Not Equal to Zero	If Z = 0, branch	REL16	3786	rrrr	6, 4	—	—	—	—	—	—	—	—
LBPL[2]	Long Branch if Plus	If N = 0, branch	REL16	378A	rrrr	6, 4	—	—	—	—	—	—	—	—
LBRA	Long Branch Always	If 1 = 1, branch	REL16	3780	rrrr	6	—	—	—	—	—	—	—	—
LBRN	Long Branch Never	If 1 = 0, branch	REL16	3781	rrrr	6	—	—	—	—	—	—	—	—
LBSR	Long Branch to Subroutine	Push (PC) (SK : SP) – 2 ⇒ SK : SP Push (CCR) (SK : SP) – 2 ⇒ SK : SP (PK : PC) + Offset ⇒ PK : PC	REL16	27F9	rrrr	10	—	—	—	—	—	—	—	—
LBVC[2]	Long Branch if Overflow Clear	If V = 0, branch	REL16	3788	rrrr	6, 4	—	—	—	—	—	—	—	—
LBVS[2]	Long Branch if Overflow Set	If V = 1, branch	REL16	3789	rrrr	6, 4	—	—	—	—	—	—	—	—
LDAA	Load A	(M) ⇒ A	IND8, X IND8, Y IND8, Z IMM8 IND16, X IND16, Y IND16, Z EXT E, X E, Y E, Z	45 55 65 75 1745 1755 1765 1775 2745 2755 2765	ff ff ff ii gggg gggg gggg hh ll — — —	6 6 6 2 6 6 6 6 6 6 6	—	—	—	—	Δ	Δ	0	—
LDAB	Load B	(M) ⇒ B	IND8, X IND8, Y IND8, Z IMM8 IND16, X IND16, Y IND16, Z EXT E, X E, Y E, Z	C5 D5 E5 F5 17C5 17D5 17E5 17F5 27C5 27D5 27E5	ff ff ff ii gggg gggg gggg hh ll — — —	6 6 6 2 6 6 6 6 6 6 6	—	—	—	—	Δ	Δ	0	Δ
LDD	Load D	(M : M + 1) ⇒ D	IND8, X IND8, Y IND8, Z IMM16 IND16, X IND16, Y IND16, Z EXT E, X E, Y E, Z	85 95 A5 37B5 37C5 37D5 37E5 37F5 2785 2795 27A5	ff ff ff jj kk gggg gggg gggg hh ll — — —	6 6 6 4 6 6 6 6 6 6 6	—	—	—	—	Δ	Δ	0	—
LDE	Load E	(M : M + 1) ⇒ E	IMM16 IND16, X IND16, Y IND16, Z EXT	3735 3745 3755 3765 3775	jj kk gggg gggg gggg hh ll	4 6 6 6 6	—	—	—	—	Δ	Δ	0	—
LDED	Load Concatenated E and D	(M : M + 1) ⇒ E (M + 2 : M + 3) ⇒ D	EXT	2771	hh ll	8	—	—	—	—	—	—	—	—

Table 47 Instruction Set Summary (Continued)

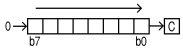
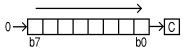
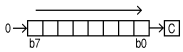
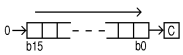
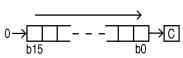
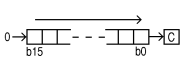
Mnemonic	Operation	Description	Address Mode	Instruction Opcode	Instruction Operand	Instruction Cycles	Condition Codes S	MV	H	EV	N	Z	V	C
LDHI	Initialize H and I	$(M : M+1)_X \Rightarrow H\ R$ $(M : M+1)_Y \Rightarrow I\ R$	EXT	27B0	—	8	—	—	—	—	—	—	—	—
LDS	Load SP	$(M : M+1) \Rightarrow SP$	IND8, X IND8, Y IND8, Z IND16, X IND16, Y IND16, Z EXT IMM16	CF DF EF 17CF 17DF 17EF 17FF 37BF	ff ff ff gggg gggg gggg hh ll jj kk	6 6 6 6 6 6 6 4	—	—	—	—	Δ	Δ	0	—
LDX	Load IX	$(M : M+1) \Rightarrow IX$	IND8, X IND8, Y IND8, Z IMM16 IND16, X IND16, Y IND16, Z EXT	CC DC EC 37BC 17CC 17DC 17EC 17FC	ff ff ff jj kk gggg gggg gggg hh ll	6 6 6 4 6 6 6 6	—	—	—	—	Δ	Δ	0	—
LDY	Load IY	$(M : M+1) \Rightarrow IY$	IND8, X IND8, Y IND8, Z IMM16 IND16, X IND16, Y IND16, Z EXT	CD DD ED 37BD 17CD 17DD 17ED 17FD	ff ff ff jj kk gggg gggg gggg hh ll	6 6 6 4 6 6 6 6	—	—	—	—	Δ	Δ	0	—
LDZ	Load IZ	$(M : M+1) \Rightarrow IZ$	IND8, X IND8, Y IND8, Z IMM16 IND16, X IND16, Y IND16, Z EXT	CE DE EE 37BE 17CE 17DE 17EE 17FE	ff ff ff jj kk gggg gggg gggg hh ll	6 6 6 4 6 6 6 6	—	—	—	—	Δ	Δ	0	—
LPSTOP	Low Power Stop	If $\overline{S}$ then STOP else NOP	INH	27F1	—	4, 20	—	—	—	—	—	—	—	—
LSR	Logical Shift Right		IND8, X IND8, Y IND8, Z IND16, X IND16, Y IND16, Z EXT	0F 1F 2F 170F 171F 172F 173F	ff ff ff gggg gggg gggg hh ll	8 8 8 8 8 8 8	—	—	—	—	0	Δ	Δ	Δ
LSRA	Logical Shift Right A		INH	370F	—	2	—	—	—	—	0	Δ	Δ	Δ
LSRB	Logical Shift Right B		INH	371F	—	2	—	—	—	—	0	Δ	Δ	Δ
LSRD	Logical Shift Right D		INH	27FF	—	2	—	—	—	—	0	Δ	Δ	Δ
LSRE	Logical Shift Right E		INH	277F	—	2	—	—	—	—	0	Δ	Δ	Δ
LSRW	Logical Shift Right Word		IND16, X IND16, Y IND16, Z EXT	270F 271F 272F 273F	gggg gggg gggg hh ll	8 8 8 8	—	—	—	—	0	Δ	Δ	Δ

Table 47 Instruction Set Summary (Continued)

Mnemonic	Operation	Description	Address Mode	Instruction Opcode	Instruction Operand	Instruction Cycles	Condition Codes S	MV	H	EV	N	Z	V	C
MAC	Multiply and Accumulate Signed 16-Bit Fractions	(HR) ∗ (IR) ⇒ E : D (AM) + (E : D) ⇒ AM Qualified (IX) ⇒ IX Qualified (IY) ⇒ IY (HR) ⇒ IZ $(M : M + 1)_X \Rightarrow HR$ $(M : M + 1)_Y \Rightarrow IR$	IMM8	7B	xoyo	12	—	Δ	—	Δ	—	—	Δ	—
MOVB	Move Byte	$(M_1) \Rightarrow M_2$	IXP to EXT EXT to IXP EXT to EXT	30 32 37FE	ff hh ll ff hh ll hh ll hh ll	8 8 10	—	—	—	—	Δ	Δ	0	—
MOVW	Move Word	$(M : M + 1_1) \Rightarrow M : M + 1_2$	IXP to EXT EXT to IXP EXT to EXT	31 33 37FF	ff hh ll ff hh ll hh ll hh ll	8 8 10	—	—	—	—	Δ	Δ	0	—
MUL	Multiply	(A) ∗ (B) ⇒ D	INH	3724	—	10	—	—	—	—	—	—	—	Δ
NEG	Negate Memory	$00 – (M) ⇒ M	IND8, X IND8, Y IND8, Z IND16, X IND16, Y IND16, Z EXT	02 12 22 1702 1712 1722 1732	ff ff ff gggg gggg gggg hh ll	8 8 8 8 8 8 8	—	—	—	—	Δ	Δ	Δ	Δ
NEGA	Negate A	$00 – (A) ⇒ A	INH	3702	—	2	—	—	—	—	Δ	Δ	Δ	Δ
NEGB	Negate B	$00 – (B) ⇒ B	INH	3712	—	2	—	—	—	—	Δ	Δ	Δ	Δ
NEGD	Negate D	$0000 – (D) ⇒ D	INH	27F2	—	2	—	—	—	—	Δ	Δ	Δ	Δ
NEGE	Negate E	$0000 – (E) ⇒ E	INH	2772	—	2	—	—	—	—	Δ	Δ	Δ	Δ
NEGW	Negate Memory Word	$0000 – (M : M + 1) ⇒ M : M + 1	IND16, X IND16, Y IND16, Z EXT	2702 2712 2722 2732	gggg gggg gggg hh ll	8 8 8 8	—	—	—	—	Δ	Δ	Δ	Δ
NOP	Null Operation	—	INH	274C	—	2	—	—	—	—	—	—	—	—
ORAA	OR A	(A) ✣ (M) ⇒ A	IND8, X IND8, Y IND8, Z IMM8 IND16, X IND16, Y IND16, Z EXT E, X E, Y E, Z	47 57 67 77 1747 1757 1767 1777 2747 2757 2767	ff ff ff ii gggg gggg gggg hh ll — — —	6 6 6 2 6 6 6 6 6 6 6	—	—	—	—	Δ	Δ	0	—
ORAB	OR B	(B) ✣ (M) ⇒ B	IND8, X IND8, Y IND8, Z IMM8 IND16, X IND16, Y IND16, Z EXT E, X E, Y E, Z	C7 D7 E7 F7 17C7 17D7 17E7 17F7 27C7 27D7 27E7	ff ff ff ii gggg gggg gggg hh ll — — —	6 6 6 2 6 6 6 6 6 6 6	—	—	—	—	Δ	Δ	0	—
ORD	OR D	(D) ✣ (M : M + 1) ⇒ D	IND8, X IND8, Y IND8, Z IMM16 IND16, X IND16, Y IND16, Z EXT E, X E, Y E, Z	87 97 A7 37B7 37C7 37D7 37E7 37F7 2787 2797 27A7	ff ff ff jj kk gggg gggg gggg hh ll — — —	6 6 6 4 6 6 6 6 6 6 6	—	—	—	—	Δ	Δ	0	—

Table 47 Instruction Set Summary (Continued)

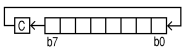
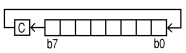
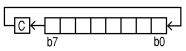
Mnemonic	Operation	Description	Address Mode	Instruction Opcode	Instruction Operand	Instruction Cycles	Condition Codes S	MV	H	EV	N	Z	V	C
ORE	OR E	(E) ✣ (M : M + 1) ⇒ E	IMM16 IND16, X IND16, Y IND16, Z EXT	3737 3747 3757 3767 3777	jj kk gggg gggg gggg hh ll	4 6 6 6 6	—	—	—	—	Δ	Δ	0	—
ORP [1]	OR Condition Code Register	(CCR) ✣ IMM16 ⇒ CCR	IMM16	373B	jj kk	4	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ
PSHA	Push A	(SK : SP) + $0001 ⇒ SK : SP Push (A) (SK : SP) – $0002 ⇒ SK : SP	INH	3708	—	4	—	—	—	—	—	—	—	—
PSHB	Push B	(SK : SP) + $0001 ⇒ SK : SP Push (B) (SK : SP) – $0002 ⇒ SK : SP	INH	3718	—	4	—	—	—	—	—	—	—	—
PSHM	Push Multiple Registers Mask bits: 0 = D 1 = E 2 = IX 3 = IY 4 = IZ 5 = K 6 = CCR 7 = (Reserved)	For mask bits 0 to 7: If mask bit set Push register (SK : SP) – 2 ⇒ SK : SP	IMM8	34	ii	4 + 2N N = number of iterations	—	—	—	—	—	—	—	—
PSHMAC	Push MAC Registers	MAC Registers ⇒ Stack	INH	27B8	—	14	—	—	—	—	—	—	—	—
PULA	Pull A	(SK : SP) + $0002 ⇒ SK : SP Pull (A) (SK : SP) – $0001 ⇒ SK : SP	INH	3709	—	6	—	—	—	—	—	—	—	—
PULB	Pull B	(SK : SP) + $0002 ⇒ SK : SP Pull (B) (SK : SP) – $0001 ⇒ SK : SP	INH	3719	—	6	—	—	—	—	—	—	—	—
PULM[1]	Pull Multiple Registers Mask bits: 0 = CCR[15:4] 1 = K 2 = IZ 3 = IY 4 = IX 5 = E 6 = D 7 = (Reserved)	For mask bits 0 to 7: If mask bit set (SK : SP) + 2 ⇒ SK : SP Pull register	IMM8	35	ii	4+2(N+1) N = number of iterations	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ
PULMAC	Pull MAC State	Stack ⇒ MAC Registers	INH	27B9	—	16	—	—	—	—	—	—	—	—
RMAC	Repeating Multiply and Accumulate Signed 16-Bit Fractions	Repeat until (E) < 0 (AM) + (H) ∗ (I) ⇒ AM Qualified (IX) ⇒ IX; Qualified (IY) ⇒ IY; $(M : M + 1)_X$ ⇒ H; $(M : M + 1)_Y$ ⇒ I (E) – 1 ⇒ E Until (E) < $0000	IMM8	FB	xoyo	6 + 12 per iteration	—	Δ	—	Δ	—	—	—	—
ROL	Rotate Left		IND8, X IND8, Y IND8, Z IND16, X IND16, Y IND16, Z EXT	0C 1C 2C 170C 171C 172C 173C	ff ff ff gggg gggg gggg hh ll	8 8 8 8 8 8 8	—	—	—	—	Δ	Δ	Δ	Δ
ROLA	Rotate Left A		INH	370C	—	2	—	—	—	—	Δ	Δ	Δ	Δ
ROLB	Rotate Left B		INH	371C	—	2	—	—	—	—	Δ	Δ	Δ	Δ

Table 47 Instruction Set Summary (Continued)

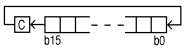
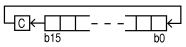
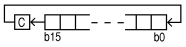
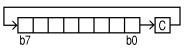
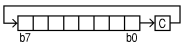
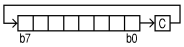
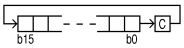
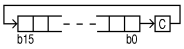
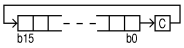
Mnemonic	Operation	Description	Address Mode	Instruction Opcode	Instruction Operand	Instruction Cycles	Condition Codes S	MV	H	EV	N	Z	V	C
ROLD	Rotate Left D		INH	27FC	—	2	—	—	—	—	Δ	Δ	Δ	Δ
ROLE	Rotate Left E		INH	277C	—	2	—	—	—	—	Δ	Δ	Δ	Δ
ROLW	Rotate Left Word		IND16, X IND16, Y IND16, Z EXT	270C 271C 272C 273C	gggg gggg gggg hh ll	8 8 8 8	—	—	—	—	Δ	Δ	Δ	Δ
ROR	Rotate Right Byte		IND8, X IND8, Y IND8, Z IND16, X IND16, Y IND16, Z EXT	0E 1E 2E 170E 171E 172E 173E	ff ff ff gggg gggg gggg hh ll	8 8 8 8 8 8 8	—	—	—	—	Δ	Δ	Δ	Δ
RORA	Rotate Right A		INH	370E	—	2	—	—	—	—	Δ	Δ	Δ	Δ
RORB	Rotate Right B		INH	371E	—	2	—	—	—	—	Δ	Δ	Δ	Δ
RORD	Rotate Right D		INH	27FE	—	2	—	—	—	—	Δ	Δ	Δ	Δ
RORE	Rotate Right E		INH	277E	—	2	—	—	—	—	Δ	Δ	Δ	Δ
RORW	Rotate Right Word		IND16, X IND16, Y IND16, Z EXT	270E 271E 272E 273E	gggg gggg gggg hh ll	8 8 8 8	—	—	—	—	Δ	Δ	Δ	Δ
RTI[3]	Return from Interrupt	(SK : SP) + 2 ⇒ SK : SP Pull CCR (SK : SP) + 2 ⇒ SK : SP Pull PC (PK : PC) – 6 ⇒ PK : PC	INH	2777	—	12	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ
RTS[4]	Return from Subroutine	(SK : SP) + 2 ⇒ SK : SP Pull PK (SK : SP) + 2 ⇒ SK : SP Pull PC (PK : PC) – 2 ⇒ PK : PC	INH	27F7	—	12	—	—	—	—	—	—	—	—
SBA	Subtract B from A	(A) – (B) ⇒ A	INH	370A	—	2	—	—	—	—	Δ	Δ	Δ	Δ
SBCA	Subtract with Carry from A	(A) – (M) – C ⇒ A	IND8, X IND8, Y IND8, Z IMM8 IND16, X IND16, Y IND16, Z EXT E, X E, Y E, Z	42 52 62 72 1742 1752 1762 1772 2742 2752 2762	ff ff ff ii gggg gggg gggg hh ll — — —	6 6 6 2 6 6 6 6 6 6 6	—	—	—	—	Δ	Δ	Δ	Δ

Table 47 Instruction Set Summary (Continued)

Mnemonic	Operation	Description	Address	Instruction			Condition Codes							
			Mode	Opcode	Operand	Cycles	S	MV	H	EV	N	Z	V	C
SBCB	Subtract with Carry from B	(B) – (M) – C ⇒ B	IND8, X	C2	ff	6	—	—	—	—	Δ	Δ	Δ	Δ
			IND8, Y	D2	ff	6								
			IND8, Z	E2	ff	6								
			IMM8	F2	ii	2								
			IND16, X	17C2	gggg	6								
			IND16, Y	17D2	gggg	6								
			IND16, Z	17E2	gggg	6								
			EXT	17F2	hh ll	6								
			E, X	27C2	—	6								
			E, Y	27D2	—	6								
			E, Z	27E2	—	6								
SBCD	Subtract with Carry from D	(D) – (M : M + 1) – C ⇒ D	IND8, X	82	ff	6	—	—	—	—	Δ	Δ	Δ	Δ
			IND8, Y	92	ff	6								
			IND8, Z	A2	ff	6								
			IMM16	37B2	jj kk	4								
			IND16, X	37C2	gggg	6								
			IND16, Y	37D2	gggg	6								
			IND16, Z	37E2	gggg	6								
			EXT	37F2	hh ll	6								
			E, X	2782	—	6								
			E, Y	2792	—	6								
			E, Z	27A2	—	6								
SBCE	Subtract with Carry from E	(E) – (M : M + 1) – C ⇒ E	IMM16	3732	jj kk	4	—	—	—	—	Δ	Δ	Δ	Δ
			IND16, X	3742	gggg	6								
			IND16, Y	3752	gggg	6								
			IND16, Z	3762	gggg	6								
			EXT	3772	hh ll	6								
SDE	Subtract D from E	(E) – (D)⇒ E	INH	2779	—	2	—	—	—	—	Δ	Δ	Δ	Δ
STAA	Store A	(A) ⇒ M	IND8, X	4A	ff	4	—	—	—	—	Δ	Δ	0	—
			IND8, Y	5A	ff	4								
			IND8, Z	6A	ff	4								
			IND16, X	174A	gggg	6								
			IND16, Y	175A	gggg	6								
			IND16, Z	176A	gggg	6								
			EXT	177A	hh ll	6								
			E, X	274A	—	4								
			E, Y	275A	—	4								
			E, Z	276A	—	4								
STAB	Store B	(B) ⇒ M	IND8, X	CA	ff	4	—	—	—	—	Δ	Δ	0	—
			IND8, Y	DA	ff	4								
			IND8, Z	EA	ff	4								
			IND16, X	17CA	gggg	6								
			IND16, Y	17DA	gggg	6								
			IND16, Z	17EA	gggg	6								
			EXT	17FA	hh ll	6								
			E, X	27CA	—	4								
			E, Y	27DA	—	4								
			E, Z	27EA	—	4								
STD	Store D	(D) ⇒ M : M + 1	IND8, X	8A	ff	4	—	—	—	—	Δ	Δ	0	—
			IND8, Y	9A	ff	4								
			IND8, Z	AA	ff	4								
			IND16, X	37CA	gggg	6								
			IND16, Y	37DA	gggg	6								
			IND16, Z	37EA	gggg	6								
			EXT	37FA	hh ll	6								
			E, X	278A	—	6								
			E, Y	279A	—	6								
			E, Z	27AA	—	6								
STE	Store E	(E) ⇒ M : M + 1	IND16, X	374A	gggg	6	—	—	—	—	Δ	Δ	0	—
			IND16, Y	375A	gggg	6								
			IND16, Z	376A	gggg	6								
			EXT	377A	hh ll	6								
STED	Store Concatenated D and E	(E) ⇒ M : M + 1 (D) ⇒ M + 2 : M + 3	EXT	2773	hh ll	8	—	—	—	—	—	—	—	—

Table 47 Instruction Set Summary (Continued)

Mnemonic	Operation	Description	Address Mode	Instruction Opcode	Instruction Operand	Instruction Cycles	Condition Codes S	MV	H	EV	N	Z	V	C
STS	Store Stack Pointer	(SP) ⇒ M : M + 1	IND8, X	8F	ff	4	—	—	—	—	Δ	Δ	0	—
			IND8, Y	9F	ff	4								
			IND8, Z	AF	ff	4								
			IND16, X	178F	gggg	6								
			IND16, Y	179F	gggg	6								
			IND16, Z	17AF	gggg	6								
			EXT	17BF	hh ll	6								
STX	Store IX	(IX) ⇒ M : M + 1	IND8, X	8C	ff	4	—	—	—	—	Δ	Δ	0	—
			IND8, Y	9C	ff	4								
			IND8, Z	AC	ff	4								
			IND16, X	178C	gggg	6								
			IND16, Y	179C	gggg	6								
			IND16, Z	17AC	gggg	6								
			EXT	17BC	hh ll	6								
STY	Store IY	(IY) ⇒ M : M + 1	IND8, X	8D	ff	4	—	—	—	—	Δ	Δ	0	—
			IND8, Y	9D	ff	4								
			IND8, Z	AD	ff	4								
			IND16, X	178D	gggg	6								
			IND16, Y	179D	gggg	6								
			IND16, Z	17AD	gggg	6								
			EXT	17BD	hh ll	6								
STZ	Store Z	(IZ) ⇒ M : M + 1	IND8, X	8E	ff	4	—	—	—	—	Δ	Δ	0	—
			IND8, Y	9E	ff	4								
			IND8, Z	AE	ff	4								
			IND16, X	178E	gggg	6								
			IND16, Y	179E	gggg	6								
			IND16, Z	17AE	gggg	6								
			EXT	17BE	hh ll	6								
SUBA	Subtract from A	(A) – (M) ⇒ A	IND8, X	40	ff	6	—	—	—	—	Δ	Δ	Δ	Δ
			IND8, Y	50	ff	6								
			IND8, Z	60	ff	6								
			IMM8	70	ii	2								
			IND16, X	1740	gggg	6								
			IND16, Y	1750	gggg	6								
			IND16, Z	1760	gggg	6								
			EXT	1770	hh ll	6								
			E, X	2740	—	6								
			E, Y	2750	—	6								
			E, Z	2760	—	6								
SUBB	Subtract from B	(B) – (M) ⇒ B	IND8, X	C0	ff	6	—	—	—	—	Δ	Δ	Δ	Δ
			IND8, Y	D0	ff	6								
			IND8, Z	E0	ff	6								
			IMM8	F0	ii	2								
			IND16, X	17C0	gggg	6								
			IND16, Y	17D0	gggg	6								
			IND16, Z	17E0	gggg	6								
			EXT	17F0	hh ll	6								
			E, X	27C0	—	6								
			E, Y	27D0	—	6								
			E, Z	27E0	—	6								
SUBD	Subtract from D	(D) – (M : M + 1) ⇒ D	IND8, X	80	ff	6	—	—	—	—	Δ	Δ	Δ	Δ
			IND8, Y	90	ff	6								
			IND8, Z	A0	ff	6								
			IMM16	37B0	jj kk	4								
			IND16, X	37C0	gggg	6								
			IND16, Y	37D0	gggg	6								
			IND16, Z	37E0	gggg	6								
			EXT	37F0	hh ll	6								
			E, X	2780	—	6								
			E, Y	2790	—	6								
			E, Z	27A0	—	6								
SUBE	Subtract from E	(E) – (M : M + 1) ⇒ E	IMM16	3730	jj kk	4	—	—	—	—	Δ	Δ	Δ	Δ
			IND16, X	3740	gggg	6								
			IND16, Y	3750	gggg	6								
			IND16, Z	3760	gggg	6								
			EXT	3770	hh ll	6								

Table 47 Instruction Set Summary (Continued)

Mnemonic	Operation	Description	Address Mode	Instruction Opcode	Instruction Operand	Instruction Cycles	Condition Codes S	MV	H	EV	N	Z	V	C
SWI	Software Interrupt	(PK : PC) + $0002 ⇒ PK : PC Push (PC) (SK : SP) – $0002 ⇒ SK : SP Push (CCR) (SK : SP) – $0002 ⇒ SK : SP $0 ⇒ PK SWI Vector ⇒ PC	INH	3720	—	16	—	—	—	—	—	—	—	—
SXT	Sign Extend B into A	If B7 = 1 then $FF ⇒ A else $00 ⇒ A	INH	27F8	—	2	—	—	—	—	Δ	Δ	—	—
TAB	Transfer A to B	(A) ⇒ B	INH	3717	—	2	—	—	—	—	Δ	Δ	0	—
TAP	Transfer A to CCR	(A[7:0]) ⇒ CCR[15:8]	INH	37FD	—	4	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ
TBA	Transfer B to A	(B) ⇒ A	INH	3707	—	2	—	—	—	—	Δ	Δ	0	—
TBEK	Transfer B to EK	(B[3:0]) ⇒ EK	INH	27FA	—	2	—	—	—	—	—	—	—	—
TBSK	Transfer B to SK	(B[3:0]) ⇒ SK	INH	379F	—	2	—	—	—	—	—	—	—	—
TBXK	Transfer B to XK	(B[3:0]) ⇒ XK	INH	379C	—	2	—	—	—	—	—	—	—	—
TBYK	Transfer B to YK	(B[3:0]) ⇒ YK	INH	379D	—	2	—	—	—	—	—	—	—	—
TBZK	Transfer B to ZK	(B[3:0]) ⇒ ZK	INH	379E	—	2	—	—	—	—	—	—	—	—
TDE	Transfer D to E	(D) ⇒ E	INH	277B	—	2	—	—	—	—	Δ	Δ	0	—
TDMSK	Transfer D to XMSK : YMSK	(D[15:8]) ⇒ X MASK (D[7:0]) ⇒ Y MASK	INH	372F	—	2	—	—	—	—	—	—	—	—
TDP[1]	Transfer D to CCR	(D) ⇒ CCR[15:4]	INH	372D	—	4	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ
TED	Transfer E to D	(E) ⇒ D	INH	27FB	—	2	—	—	—	—	Δ	Δ	0	—
TEDM	Transfer E and D to AM[31:0] Sign Extend AM	(E) ⇒ AM[31:16] (D) ⇒ AM[15:0] AM[35:32] = AM31	INH	27B1	—	4	—	0	—	0	—	—	—	—
TEKB	Transfer EK to B	(EK) ⇒ B[3:0] $0 ⇒ B[7:4]	INH	27BB	—	2	—	—	—	—	—	—	—	—
TEM	Transfer E to AM[31:16] Sign Extend AM Clear AM LSB	(E) ⇒ AM[31:16] $00 ⇒ AM[15:0] AM[35:32] = AM31	INH	27B2	—	4	—	0	—	0	—	—	—	—
TMER	Transfer Rounded AM to E	Rounded (AM) ⇒ Temp If (SM • (EV ✣ MV)) then Saturation Value ⇒ E else Temp[31:16] ⇒ E	INH	27B4	—	6	—	Δ	—	Δ	Δ	Δ	—	—
TMET	Transfer Truncated AM to E	If (SM • (EV ✣ MV)) then Saturation Value ⇒ E else AM[31:16] ⇒ E	INH	27B5	—	2	—	—	—	—	Δ	Δ	—	—
TMXED	Transfer AM to IX : E : D	AM[35:32] ⇒ IX[3:0] AM35 ⇒ IX[15:4] AM[31:16] ⇒ E AM[15:0] ⇒ D	INH	27B3	—	6	—	—	—	—	—	—	—	—
TPA	Transfer CCR to A	(CCR[15:8]) ⇒ A	INH	37FC	—	2	—	—	—	—	—	—	—	—
TPD	Transfer CCR to D	(CCR) ⇒ D	INH	372C	—	2	—	—	—	—	—	—	—	—
TSKB	Transfer SK to B	(SK) ⇒ B[3:0] $0 ⇒ B[7:4]	INH	37AF	—	2	—	—	—	—	—	—	—	—
TST	Test Byte Zero or Minus	(M) – $00	IND8, X IND8, Y IND8, Z IND16, X IND16, Y IND16, Z EXT	06 16 26 1706 1716 1726 1736	ff ff ff gggg gggg gggg hh ll	6 6 6 6 6 6 6	—	—	—	—	Δ	Δ	0	0
TSTA	Test A for Zero or Minus	(A) – $00	INH	3706	—	2	—	—	—	—	Δ	Δ	0	0
TSTB	Test B for Zero or Minus	(B) – $00	INH	3716	—	2	—	—	—	—	Δ	Δ	0	0
TSTD	Test D for Zero or Minus	(D) – $0000	INH	27F6	—	2	—	—	—	—	Δ	Δ	0	0
TSTE	Test E for Zero or Minus	(E) – $0000	INH	2776	—	2	—	—	—	—	Δ	Δ	0	0

Table 47 Instruction Set Summary (Continued)

Mnemonic	Operation	Description	Address Mode	Instruction Opcode	Instruction Operand	Instruction Cycles	Condition Codes S	MV	H	EV	N	Z	V	C
TSTW	Test for Zero or Minus Word	(M : M + 1) – $0000	IND16, X IND16, Y IND16, Z EXT	2706 2716 2726 2736	gggg gggg gggg hh ll	6 6 6 6	—	—	—	—	Δ	Δ	0	0
TSX	Transfer SP to IX	(SK : SP) + $0002 ⇒ XK : IX	INH	274F	—	2	—	—	—	—	—	—	—	—
TSY	Transfer SP to IY	(SK : SP) + $0002 ⇒ YK : IY	INH	275F	—	2	—	—	—	—	—	—	—	—
TSZ	Transfer SP to IZ	(SK : SP) + $0002 ⇒ ZK : IZ	INH	276F	—	2	—	—	—	—	—	—	—	—
TXKB	Transfer XK to B	(XK) ⇒ B[3:0] $0 ⇒ B[7:4]	INH	37AC	—	2	—	—	—	—	—	—	—	—
TXS	Transfer IX to SP	(XK : IX) – $0002 ⇒ SK : SP	INH	374E	—	2	—	—	—	—	—	—	—	—
TXY	Transfer IX to IY	(XK : IX) ⇒ YK : IY	INH	275C	—	2	—	—	—	—	—	—	—	—
TXZ	Transfer IX to IZ	(XK : IX) ⇒ ZK : IZ	INH	276C	—	2	—	—	—	—	—	—	—	—
TYKB	Transfer YK to B	(YK) ⇒ B[3:0] $0 ⇒ B[7:4]	INH	37AD	—	2	—	—	—	—	—	—	—	—
TYS	Transfer IY to SP	(YK : IY) – $0002 ⇒ SK : SP	INH	375E	—	2	—	—	—	—	—	—	—	—
TYX	Transfer IY to IX	(YK : IY) ⇒ XK : IX	INH	274D	—	2	—	—	—	—	—	—	—	—
TYZ	Transfer IY to IZ	(YK : IY) ⇒ ZK : IZ	INH	276D	—	2	—	—	—	—	—	—	—	—
TZKB	Transfer ZK to B	(ZK) ⇒ B[3:0] $0 ⇒ B[7:4]	INH	37AE	—	2	—	—	—	—	—	—	—	—
TZS	Transfer IZ to SP	(ZK : IZ) – $0002 ⇒ SK : SP	INH	376E	—	2	—	—	—	—	—	—	—	—
TZX	Transfer IZ to IX	(ZK : IZ) ⇒ XK : IX	INH	274E	—	2	—	—	—	—	—	—	—	—
TZY	Transfer IZ to IY	(ZK : IZ) ⇒ ZK : IY	INH	275E	—	2	—	—	—	—	—	—	—	—
WAI	Wait for Interrupt	WAIT	INH	27F3	—	8	—	—	—	—	—	—	—	—
XGAB	Exchange A with B	(A) ⇔ (B)	INH	371A	—	2	—	—	—	—	—	—	—	—
XGDE	Exchange D with E	(D) ⇔ (E)	INH	277A	—	2	—	—	—	—	—	—	—	—
XGDX	Exchange D with IX	(D) ⇔ (IX)	INH	37CC	—	2	—	—	—	—	—	—	—	—
XGDY	Exchange D with IY	(D) ⇔ (IY)	INH	37DC	—	2	—	—	—	—	—	—	—	—
XGDZ	Exchange D with IZ	(D) ⇔ (IZ)	INH	37EC	—	2	—	—	—	—	—	—	—	—
XGEX	Exchange E with IX	(E) ⇔ (IX)	INH	374C	—	2	—	—	—	—	—	—	—	—
XGEY	Exchange E with IY	(E) ⇔ (IY)	INH	375C	—	2	—	—	—	—	—	—	—	—
XGEZ	Exchange E with IZ	(E) ⇔ (IZ)	INH	376C	—	2	—	—	—	—	—	—	—	—

1. CCR[15:4] change according to results of operation. The PK field is not affected.
2. Cycle times for conditional branches are shown in "taken, not taken" order.
3. CCR[15:0] change according to copy of CCR pulled from stack.
4. PK field changes according to state pulled from stack. The rest of the CCR is not affected.

Table 48 Instruction Set Abbreviations and Symbols

A	—	Accumulator A
AM	—	Accumulator M
B	—	Accumulator B
CCR	—	Condition code register
D	—	Accumulator D
E	—	Accumulator E
EK	—	Extended addressing extension field
IR	—	MAC multiplicand register
HR	—	MAC multiplier register
IX	—	Index register X
IY	—	Index register Y
IZ	—	Index register Z
K	—	Address extension register
PC	—	Program counter
PK	—	Program counter extension field
SK	—	Stack pointer extension field
SL	—	Multiply and accumulate sign latch
SP	—	Stack pointer
XK	—	Index register X extension field
YK	—	Index register Y extension field
ZK	—	Index register Z extension field
XMSK	—	Modulo addressing index register X mask
YMSK	—	Modulo addressing index register Y mask
S	—	Stop disable control bit
MV	—	AM overflow indicator
H	—	Half carry indicator
EV	—	AM extended overflow indicator
N	—	Negative indicator
Z	—	Zero indicator
V	—	Two's complement overflow indicator
C	—	Carry/borrow indicator
IP	—	Interrupt priority field
SM	—	Saturation mode control bit
PK	—	Program counter extension field
—	—	Bit not affected
Δ	—	Bit changes as specified
0	—	Bit cleared
1	—	Bit set
M	—	Memory location used in operation
R	—	Result of operation
S	—	Source data

X	—	Register used in operation
M	—	Address of one memory byte
M +1	—	Address of byte at M + $0001
M : M + 1	—	Address of one memory word
(…)X	—	Contents of address pointed to by IX
(...)Y	—	Contents of address pointed to by IY
(...)Z	—	Contents of address pointed to by IZ
E, X	—	IX with E offset
E, Y	—	IY with E offset
E, Z	—	IZ with E offset
EXT	—	Extended
EXT20	—	20-bit extended
IMM8	—	8-bit immediate
IMM16	—	16-bit immediate
IND8, X	—	IX with unsigned 8-bit offset
IND8, Y	—	IY with unsigned 8-bit offset
IND8, Z	—	IZ with unsigned 8-bit offset
IND16, X	—	IX with signed 16-bit offset
IND16, Y	—	IY with signed 16-bit offset
IND16, Z	—	IZ with signed 16-bit offset
IND20, X	—	IX with signed 20-bit offset
IND20, Y	—	IY with signed 20-bit offset
IND20, Z	—	IZ with signed 20-bit offset
INH	—	Inherent
IXP	—	Post-modified indexed
REL8	—	8-bit relative
REL16	—	16-bit relative
b	—	4-bit address extension
ff	—	8-bit unsigned offset
gggg	—	16-bit signed offset
hh	—	High byte of 16-bit extended address
ii	—	8-bit immediate data
jj	—	High byte of 16-bit immediate data
kk	—	Low byte of 16-bit immediate data
ll	—	Low byte of 16-bit extended address
mm	—	8-bit mask
mmmm	—	16-bit mask
rr	—	8-bit unsigned relative offset
rrrr	—	16-bit signed relative offset
xo	—	MAC index register X offset
yo	—	MAC index register Y offset
z	—	4-bit zero extension

+	—	Addition
−	—	Subtraction or negation (two's complement)
*	—	Multiplication
/	—	Division
>	—	Greater
<	—	Less
=	—	Equal
≥	—	Equal or greater
≤	—	Equal or less
≠	—	Not equal

•	—	AND
✣	—	Inclusive OR (OR)
⊕	—	Exclusive OR (EOR)
$\overline{\text{NOT}}$	—	Complementation
:	—	Concatenation
⇒	—	Transferred
⇔	—	Exchanged
±	—	Sign bit; also used to show tolerance
«	—	Sign extension
%	—	Binary value
$	—	Hexadecimal value

4.7 Exceptions

An exception is an event that preempts normal instruction process. Exception processing makes the transition from normal instruction execution to execution of a routine that deals with an exception.

Each exception has an assigned vector that points to an associated handler routine. Exception processing includes all operations required to transfer control to a handler routine, but does not include execution of the handler routine itself. Keep the distinction between exception processing and execution of an exception handler in mind while reading this section.

4.7.1 Exception Vectors

An exception vector is the address of a routine that handles an exception. Exception vectors are contained in a data structure called the exception vector table, which is located in the first 512 bytes of bank 0. Refer to **Table 49** for the exception vector table.

All vectors except the reset vector consist of one word and reside in data space. The reset vector consists of four words that reside in program space. There are 52 predefined or reserved vectors, and 200 user-defined vectors.

Each vector is assigned an 8-bit number. Vector numbers for some exceptions are generated by external devices; others are supplied by the processor. There is a direct mapping of vector number to vector table address. The processor left shifts the vector number one place (multiplies by two) to convert it to an address.

Table 49 Exception Vector Table

Vector Number	Vector Address	Address Space	Type of Exception
0	0000	P	Reset — Initial ZK, SK, and PK
	0002	P	Reset — Initial PC
	0004	P	Reset — Initial SP
	0006	P	Reset — Initial IZ (Direct Page)
4	0008	D	Breakpoint
5	000A	D	Bus Error
6	000C	D	Software Interrupt
7	000E	D	Illegal Instruction
8	0010	D	Division by Zero
9 – E	0012 – 001C	D	Unassigned, Reserved
F	001E	D	Uninitialized Interrupt
10	0020	D	Unassigned, Reserved
11	0022	D	Level 1 Interrupt Autovector
12	0024	D	Level 2 Interrupt Autovector
13	0026	D	Level 3 Interrupt Autovector
14	0028	D	Level 4 Interrupt Autovector
15	002A	D	Level 5 Interrupt Autovector
16	002C	D	Level 6 Interrupt Autovector
17	002E	D	Level 7 Interrupt Autovector
18	0030	D	Spurious Interrupt
19 – 37	0032 – 006E	D	Unassigned, Reserved
38 – FF	0070 – 01FE	D	User-Defined Interrupts

4.7.2 Exception Stack Frame

During exception processing, the contents of the program counter and condition code register are stacked at a location pointed to by SK : SP. Unless it is altered during exception processing, the stacked PK : PC value is the address of the next instruction in the current instruction stream, plus $0006. **Figure 16** shows the exception stack frame.

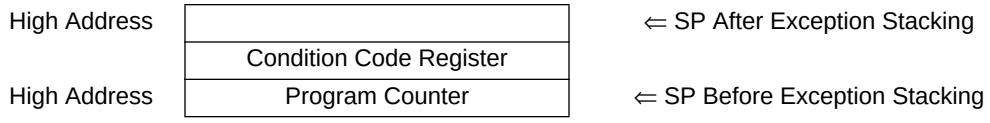


Figure 16 Exception Stack Frame Format

4.7.3 Exception Processing Sequence

Exception processing is performed in four phases.

A. Priority of all pending exceptions is evaluated, and the highest priority exception is processed first.
B. Processor state is stacked, then the CCR PK extension field is cleared.
C. An exception vector number is acquired and converted to a vector address.
D. The content of the vector address is loaded into the PC, and the processor jumps to the exception handler routine.

There are variations within each phase for differing types of exceptions. However, all vectors but the reset vectors contain 16-bit addresses, and the PK field is cleared. Exception handlers must be located within bank 0 or vectors must point to a jump table.

4.7.4 Types of Exceptions

Exceptions can be either internally or externally generated. External exceptions, which are defined as asynchronous, include interrupts, bus errors ($\overline{\text{BERR}}$), breakpoints ($\overline{\text{BKPT}}$), and resets ($\overline{\text{RESET}}$). Internal exceptions, which are defined as synchronous, include the software interrupt (SWI) instruction, the background (BGND) instruction, illegal instruction exceptions, and the divide-by-zero exception.

4.7.4.1 Asynchronous Exceptions

Asynchronous exceptions occur without reference to CPU16 or IMB clocks, but exception processing is synchronized. For all asynchronous exceptions but $\overline{\text{RESET}}$, exception processing begins at the first instruction boundary following recognition of an exception.

Because of pipelining, the stacked return PK : PC value for all asynchronous exceptions, other than reset, is equal to the address of the next instruction in the current instruction stream plus $0006. The RTI instruction, which must terminate all exception handler routines, subtracts $0006 from the stacked value to resume execution of the interrupted instruction stream.

4.7.4.2 Synchronous Exceptions

Synchronous exception processing is part of an instruction definition. Exception processing for synchronous exceptions is always completed, and the first instruction of the handler routine is always executed, before interrupts are detected.

Because of pipelining, the value of PK : PC at the time a synchronous exception executes is equal to the address of the instruction that causes the exception plus $0006. Because RTI always subtracts $0006 upon return, the stacked PK : PC must be adjusted by the instruction that caused the exception so that execution resumes with the following instruction. For this reason, $0002 is added to the PK : PC value before it is stacked.

4.7.5 Multiple Exceptions

Each exception has a hardware priority based upon its relative importance to system operation. Asynchronous exceptions have higher priorities than synchronous exceptions. Exception processing for multiple exceptions is completed by priority, from highest to lowest. Priority governs the order in which exception processing occurs, not the order in which exception handlers are executed.

Unless a bus error, a breakpoint, or a reset occurs during exception processing, the first instruction of all exception handler routines is guaranteed to execute before another exception is processed. Because interrupt exceptions have higher priority than synchronous exceptions, the first instruction in an interrupt handler is executed before other interrupts are sensed.

Bus error, breakpoint, and reset exceptions that occur during exception processing of a previous exception are processed before the first instruction of that exception's handler routine. The converse is not true. If an interrupt occurs during bus error exception processing, for example, the first instruction of the exception handler is executed before interrupts are sensed. This permits the exception handler to mask interrupts during execution.

4.7.6 RTI Instruction

The return-from-interrupt instruction (RTI) must be the last instruction in all exception handlers except the RESET handler. RTI pulls the exception stack frame that was pushed onto the system stack during exception processing, and restores processor state. Normal program flow resumes at the address of the instruction that follows the last instruction executed before exception processing began.

RTI is not used in the RESET handler because $\overline{\text{RESET}}$ initializes the stack pointer and does not create a stack frame.

5 Queued Serial Module

The QSM contains two serial interfaces: the queued serial peripheral interface (QSPI) and the serial communication interface (SCI). **Figure 17** shows the QSM block diagram.

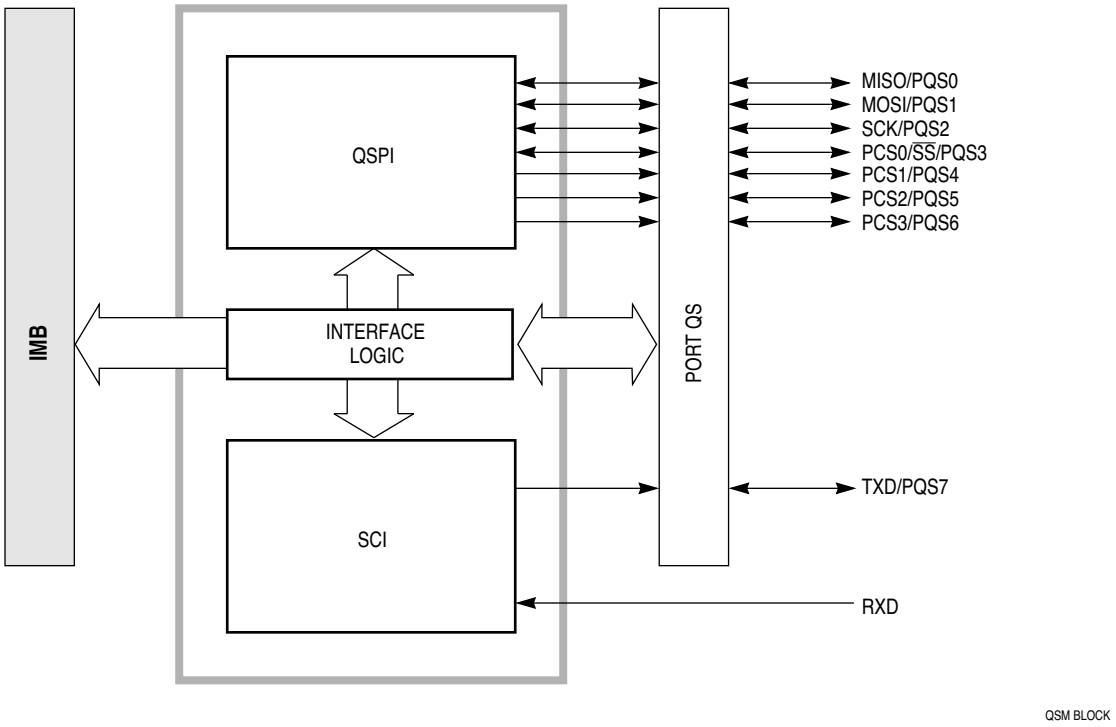


Figure 17 QSM Block Diagram

5.1 Overview

The QSPI provides peripheral expansion or interprocessor communication through a full-duplex, synchronous, three-line bus: data in, data out, and a serial clock. Four programmable peripheral chip-select pins provide addressability for up to 16 peripheral devices. A self-contained RAM queue allows up to 16 serial transfers of 8 to 16 bits each, or transmission of a 256-bit data stream without CPU intervention. A special wraparound mode supports continuous sampling of a serial peripheral, with automatic QSPI RAM updating, which makes the interface to A/D converters more efficient.

The SCI provides a standard non-return to zero (NRZ) mark/space format. It operates in either full- or half-duplex mode. There are separate transmitter and receiver enable bits and dual data buffers. A modulus-type baud rate generator provides rates from 110 to 655 kbaud with a 20.97 MHz system clock. Word length of either 8 or 9 bits is software selectable. Optional parity generation and detection provide either even or odd parity check capability. Advanced error detection circuitry catches glitches of up to 1/16 of a bit time in duration. Wake-up functions allow the CPU to run uninterrupted until meaningful data is available.

5.2 Address Map

Table 50 shows the QSM address map.

Table 50 QSM Address Map

Address	15 8	7 0
$YFFC00[1]	QSM MODULE CONFIGURATION (QSMCR)	
$YFFC02	QSM TEST (QTEST)	
$YFFC04	QSM INTERRUPT LEVEL (QILR)	QSM INTERRUPT VECTOR (QIVR)
$YFFC06	RESERVED	
$YFFC08	SCI CONTROL 0 REGISTER (SCCR0)	
$YFFC0A	SCI CONTROL 1 REGISTER (SCCR1)	
$YFFC0C	SCI STATUS REGISTER (SCSR)	
$YFFC0E	SCI DATA REGISTER (SCDR)	
$YFFC10	RESERVED	
$YFFC12	RESERVED	
$YFFC14	NOT USED	PQS DATA (PORTQS)
$YFFC16	PQS PIN ASSIGNMENT (PQSPAR)	PQS DATA DIRECTION (DDRQS)
$YFFC18	SPI CONTROL 0 REGISTER (SPCR0)	
$YFFC1A	SPI CONTROL 1 REGISTER (SPCR1)	
$YFFC1C	SPI CONTROL 2 REGISTER (SPCR2)	
$YFFC1E	SPI CONTROL 3 REGISTER (SPCR3)	SPI STATUS REGISTER (SPSR)
$YFFC20 – $YFFCFF	RESERVED	
$YFFD00 – $YFFD1F	RECEIVE RAM (RR[0:F])	
$YFFD20 – $YFFD3F	TRANSMIT RAM (TR[0:F])	
$YFFD40 – $YFFD4F	COMMAND RAM (CR[0:F])	

1. Y = M111, where M is the logic state of the MM bit in the SCIMCR.

5.3 Pin Function

Table 51 is a summary of the functions of the QSM pins when they are not configured for general-purpose I/O. The QSM data direction register (DDRQS) designates each pin except RXD as an input or output.

Table 51 QSM Pin Functions

Pin	Mode	Pin Function
MISO	Master	Serial Data Input to QSPI
	Slave	Serial Data Output from QSPI
MOSI	Master	Serial Data Output from QSPI
	Slave	Serial Data Input to QSPI
SCK	Master	Clock Output from QSPI
	Slave	Clock Input to QSPI
PCS0/$\overline{SS}$	Master	Input: Assertion Causes Mode Fault Output: Selects Peripherals
	Slave	Input: Selects the QSPI
PCS[3:1]	Master	Output: Selects Peripherals
	Slave	None
TXD	Transmit	Serial Data Output from SCI
RXD	Receive	Serial Data Input to SCI

5.4 QSM Registers

QSM registers are divided into four categories: QSM global registers, QSM pin control registers, QSPI submodule registers, and SCI submodule registers. The QSPI and SCI registers are defined in separate sections below. Writes to unimplemented register bits have no meaning or effect, and reads from unimplemented bits always return a logic zero value.

5.4.1 Global Registers

The QSM global registers contain system parameters used by both the QSPI and the SCI submodules. These registers contain the bits and fields used to configure the QSM.

QSMCR — 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[3:0]			
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The QSMCR contains parameters for the QSM/CPU/intermodule bus (IMB) interface.

STOP — Stop Enable

0 = Normal QSM clock operation

1 = QSM clock operation stopped

STOP places the QSM in a low-power state by disabling the system clock in most parts of the module. The QSMCR is the only register guaranteed to be readable while STOP is asserted. The QSPI RAM is not readable. However, writes to RAM or any register are guaranteed to be valid while STOP is asserted. STOP can be negated by the CPU and by reset.

The system software must stop each submodule before asserting STOP to avoid complications at restart and to avoid data corruption. The SCI submodule receiver and transmitter should be disabled, and the operation should be verified for completion before asserting STOP. The QSPI submodule should be stopped by asserting the HALT bit in SPCR3 and by asserting STOP after the HALTA flag is set.

FRZ1 — FREEZE Assertion Response

0 = Ignore the FREEZE signal on the IMB

1 = Halt the QSPI (on a transfer boundary)

FRZ1 determines what action is taken by the QSPI when the FREEZE signal of the IMB is asserted. FREEZE is asserted whenever the CPU enters the background mode.

FRZ0 — Not Implemented

Bits [12:8] — Not Implemented

SUPV — Supervisor/Unrestricted Data Space

This bit has no effect because the CPU16 always operates in the supervisor mode.

IARB[3:0] — Interrupt Arbitration Identification Number

The IARB field is used to arbitrate between simultaneous interrupt requests of the same priority. Each module that can generate interrupt requests must be assigned a unique, non-zero IARB field value.

QTEST — QSM Test Register **$YFFC02**

QTEST is used during factory testing of the QSM.

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	ILQSPI[2:0]			ILSCI[2:0]			QIVR							
RESET:															
0	0	0	0	0	0	0	0								

QILR determines the priority level of interrupts requested by the QSM.

ILQSPI[2:0] — Interrupt Level for QSPI

ILQSPI determines the priority of QSPI interrupts. This field must be given a value between $0 (interrupts disabled) to $7 (highest priority).

ILSCI[2:0] — Interrupt Level of SCI

ILSCI determines the priority of SCI interrupts. This field must be given a value between $0 (interrupts disabled) to $7 (highest priority).

If ILQSPI and ILSCI are the same nonzero value, and both submodules simultaneously request interrupt service, QSPI has priority.

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
QILR								INTV[7:0]							
RESET:															
								0	0	0	0	1	1	1	1

At reset, QIVR is initialized to $0F, which corresponds to the uninitialized interrupt vector in the exception table. This vector is selected until QIVR is written. A user-defined vector ($40–$FF) should be written to QIVR during QSM initialization.

After initialization, QIVR determines which two vectors in the exception vector table are to be used for QSM interrupts. The QSPI and SCI submodules have separate interrupt vectors adjacent to each other. Both submodules use the same interrupt vector with the least significant bit (LSB) determined by the submodule causing the interrupt.

The value of INTV0 used during an interrupt-acknowledge cycle is supplied by the QSM. During an interrupt-acknowledge cycle, INTV[7:1] are driven on DATA[7:1] IMB lines. DATA0 is negated for an SCI interrupt and asserted for a QSPI interrupt. Writes to INTV0 have no meaning or effect. Reads of INTV0 return a value of one.

5.4.2 Pin Control Registers

The QSM uses nine pins, eight of which form a parallel port (PORTQS) on the MCU. Although these pins are used by the serial subsystems, any pin can alternately be assigned as general-purpose I/O on a pin-by-pin basis.

Pins used for general-purpose I/O must not be assigned to the QSPI by register PQSPAR. To avoid driving incorrect data, the first byte to be output must be written before DDRQS is configured. DDRQS must then be written to determine the direction of data flow and to output the value contained in register PORTQS. Subsequent data for output is written to PORTQS.

PORTQS — Port QS Data Register **$YFFC14**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NOT USED								PQS7	PQS6	PQS5	PQS4	PQS3	PQS2	PQS1	PQS0
RESET:								0	0	0	0	0	0	0	0

PORTQS latches I/O data. Writes drive pins defined as outputs. Reads return data present on the pins. To avoid driving undefined data, first write a byte to PORTQS, then configure DDRQS.

PQSPAR — PORT QS Pin Assignment Register **$YFFC16**
DDRQS — PORT QS Data Direction Register **$YFFC17**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	PQSPA6	PQSPA5	PQSPA4	PQSPA3	0	PQSPA1	PQSPA0	DDQS7	DDQS6	DDQS5	DDQS4	DDQS3	DDQS2	DDQS1	DDQS0
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Clearing a bit in PQSPAR assigns the corresponding pin to general-purpose I/O; setting a bit assigns the pin to the QSPI. PQSPAR does not affect operation of the SCI. **Table 52** displays PQSPAR pin assignments.

Table 52 PQSPAR Pin Assignments

PQSPAR Field	PQSPAR Bit	Pin Function
PQSPA0	0	PQS0
	1	MISO
PQSPA1	0	PQS1
	1	MOSI
—	—	PQS2[1]
	—	SCK
PQSPA3	0	PQS3
	1	PCS0/$\overline{SS}$
PQSPA4	0	PQS4
	1	PCS1
PQSPA5	0	PQS5
	1	PCS2
PQSPA6	0	PQS6
	1	PCS3
—	—	PQS7[2]
	—	TXD

1. PQS2 is a digital I/O pin unless the SPI is enabled (SPE in SPCR1 set), in which case it becomes SPI serial clock SCK.
2. PQS7 is a digital I/O pin unless the SCI transmitter is enabled (TE in SCCR1 = 1), in which case it becomes SCI serial output TXD.

DDRQS determines whether pins are inputs or outputs. Clearing a bit makes the corresponding pin an input; setting a bit makes the pin an output. DDRQS affects both QSPI function and I/O function. **Table 53** shows the effect of DDRQS on QSM pin functions.

Table 53 Effect of DDRQS on QSM Pin Function

QSM Pin	Mode	DDRQS Bit	Bit State	Pin Function
MISO	Master	DDQS0	0	Serial Data Input to QSPI
			1	Disables Data Input
	Slave		0	Disables Data Output
			1	Serial Data Output from QSPI
MOSI	Master	DDQS1	0	Disables Data Output
			1	Serial Data Output from QSPI
	Slave		0	Serial Data Input to QSPI
			1	Disables Data Input
SCK[1]	Master	DDQS2	0	Disables Clock Output
			1	Clock Output from QSPI
	Slave		0	Clock Input to QSPI
			1	Disables Clock Input
PCS0/$\overline{SS}$	Master	DDQS3	0	Assertion Causes Mode Fault
			1	Chip-Select Output
	Slave		0	QSPI Slave Select Input
			1	Disables Select Input
PCS[3:1]	Master	DDQS[4:6]	0	Disables Chip-Select Output
			1	Chip-Select Output
	Slave		0	Inactive
			1	Inactive
TXD[2]	Transmit	DDQS7	X	Serial Data Output from SCI
RXD	Receive	None	NA	Serial Data Input to SCI

1. PQS2 is a digital I/O pin unless the SPI is enabled (SPE in SPCR1 set), in which case it becomes SPI serial clock SCK.
2. PQS7 is a digital I/O pin unless the SCI transmitter is enabled (TE in SCCR1 = 1), in which case it becomes SCI serial output TXD.

DDRQS determines the direction of the TXD pin only when the SCI transmitter is disabled. When the SCI transmitter is enabled, the TXD pin is an output.

5.5 QSPI Submodule

The QSPI submodule communicates with external devices through a synchronous serial bus. The QSPI is fully compatible with the serial peripheral interface (SPI) systems found on other Motorola products. **Figure 18** shows a block diagram of the QSPI.

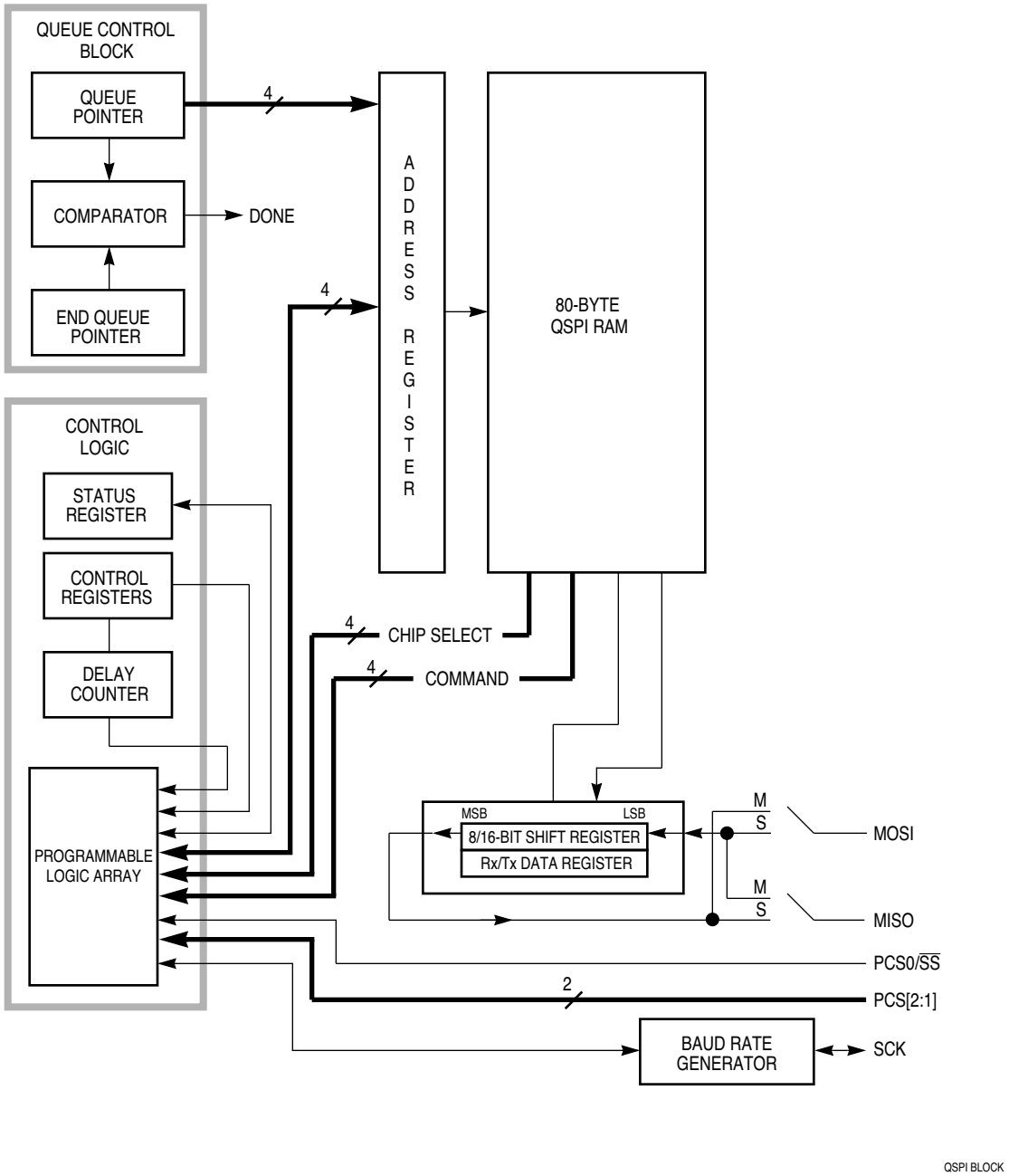


Figure 18 QSPI Block Diagram

5.5.1 QSPI Pins

Seven pins are associated with the QSPI. When not needed for a QSPI function, they can be configured as general-purpose I/O pins. The PCS0/$\overline{SS}$ pin can function as a peripheral chip-select output, slave select input, or general-purpose I/O. Refer to **Table 54** for QSPI input and output pins and their functions.

Table 54 QSPI Pins

Pin Name(s)	Mnemonic(s)	Mode	Function
Master In Slave Out	MISO	Master Slave	Serial Data Input to QSPI Serial Data Output from QSPI
Master Out Slave In	MOSI	Master Slave	Serial Data Output from QSPI Serial Data Input to QSPI
Serial Clock	SCK	Master Slave	Clock Output from QSPI Clock Input to QSPI
Peripheral Chip Selects	PCS[3:1]	Master	Select Peripherals
Peripheral Chip Select Slave Select	PCS0 $\overline{SS}$	Master Master Slave	Selects Peripheral Causes mode fault Initiates Serial Transfer

5.5.2 QSPI Registers

The programming model for the QSPI submodule consists of the QSM global and pin control registers, four QSPI control registers, one status register, and the 80-byte QSPI RAM.

The CPU can read and write to registers and RAM. The four control registers must be initialized before the QSPI is enabled to ensure defined operation. SPCR1 should be written last because it contains QSPI enable bit SPE. Asserting this bit starts the QSPI. The QSPI control registers are reset to a defined state and can then be changed by the CPU. Reset values are shown below each register.

Table 55 shows a memory map of the QSPI.

Table 55 QSPI Memory Map

Address	Name	Usage
$YFFC18	SPCR0	QSPI Control Register 0
$YFFC1A	SPCR1	QSPI Control Register 1
$YFFC1C	SPCR2	QSPI Control Register 2
$YFFC1E	SPCR3	QSPI Control Register 3
$YFFC1F	SPSR	QSPI Status Register
$YFFD00	RR[0:F]	QSPI Receive Data (16 Words)
$YFFD20	TR[0:F]	QSPI Transmit Data (16 Words)
$YFFD40	CR[0:F]	QSPI Command Control (8 Words)

Writing a different value into any control register except SPCR2 while the QSPI is enabled disrupts operation. SPCR2 is buffered to prevent disruption of the current serial transfer. After completion of the current serial transfer, the new SPCR2 values become effective.

Writing the same value into any control register except SPCR2 while the QSPI is enabled has no effect on QSPI operation. Rewriting NEWQP[3:0] in SPCR2 causes execution to restart at the designated location.

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[3:0]				CPOL	CPHA	SPBR[7:0]							
RESET:															
0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0

SPCR0 contains parameters for configuring the QSPI before it is enabled. The CPU can read and write this register. The QSM has read-only access.

MSTR — Master/Slave Mode Select

0 = QSPI is a slave device and only responds to externally generated serial data.

1 = QSPI is system master and can initiate transmission to external SPI devices.

MSTR configures the QSPI for either master or slave mode operation. This bit is cleared on reset and may only be written by the CPU.

WOMQ — Wired-OR Mode for QSPI Pins

0 = Outputs have normal MOS drivers.

1 = Pins designated for output by DDRQS have open-drain drivers.

WOMQ allows the wired-OR function to be used on QSPI pins, regardless of whether they are used as general-purpose outputs or as QSPI outputs. WOMQ affects the QSPI pins regardless of whether the QSPI is enabled or disabled.

BITS[3:0] — Bits Per Transfer

In master mode, when BITSE in a command is set, the BITS[3:0] field determines the number of data bits transferred. When BITSE is cleared, eight bits are transferred. Reserved values default to eight bits. In slave mode, the command RAM is not used and the setting of BITSE has no effect on QSPI transfers. Instead, the BITS[3:0] field determines the number of bits the QSPI will receive during each transfer before storing the received data.

Table 56 shows the number of bits per transfer.

Table 56 Bits Per Transfer

BITS[3:0]	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

0 = The inactive state value of SCK is logic level zero.
1 = The inactive state value of SCK is logic level one.

CPOL is used to determine the inactive state value of the serial clock (SCK). It is used with CPHA to produce a desired clock/data relationship between master and slave devices.

CPHA — Clock Phase

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

CPHA determines which edge of SCK causes data to change and which edge causes data to be captured. CPHA is used with CPOL to produce a desired clock/data relationship between master and slave devices. CPHA is set at reset.

SPBR[7:0] — Serial Clock Baud Rate

The QSPI uses a modulus counter to derive SCK baud rate from the MCU system clock. Baud rate is selected by writing a value from 2 to 255 into the SPBR[7:0] field. The following equation determines the SCK baud rate:

$$\text{SCK Baud Rate} = \frac{\text{System Clock}}{2 \times \text{SPBR[7:0]}}$$

or

$$\text{SPBR[7:0]} = \frac{\text{System Clock}}{2 \times \text{SCK Baud Rate Desired}}$$

Giving SPBR[7:0] a value of zero or one disables the baud rate generator. SCK is disabled and assumes its inactive state value. No serial transfers occur. At reset, baud rate is initialized to one eighth of the system clock frequency.

SPRC1 — QSPI Control Register 1 **$YFFC1A**

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

SPCR1 contains parameters for configuring the QSPI before it is enabled. The CPU can read and write this register, but the QSM has read access only, except for SPE, which is automatically cleared by the QSPI after completing all serial transfers, or when a mode fault occurs.

SPE — QSPI Enable

0 = QSPI is disabled. QSPI pins can be used for general-purpose I/O.
1 = QSPI is enabled. Pins allocated by PQSPAR are controlled by the QSPI.

DSCKL[6:0] — Delay before SCK

When the DSCK bit in command RAM is set, this field determines the length of delay from PCS valid to SCK transition. PCS can be any of the four peripheral chip-select pins. The following equation determines the actual delay before SCK:

$$\text{PCS to SCK Delay} = \frac{\text{DSCKL[6:0]}}{\text{System Clock}}$$

where DSCKL[6:0] equals {1, 2, 3,..., 127}.

When the DSCK value of a queue entry equals zero, then DSCKL[6:0] is not used. Instead, the PCS valid-to-SCK transition is one-half SCK period.

DTL[7:0] — Length of Delay after Transfer

When the DT bit in command RAM is set, this field determines the length of delay after serial transfer. The following equation is used to calculate the delay:

$$\text{Delay after Transfer} = \frac{32 \times \text{DTL[7:0]}}{\text{System Clock}}$$

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

A zero value for DTL[7:0] causes a delay-after-transfer value of 8192/System Clock.

If DT equals zero, a standard delay is inserted.

$$\text{Standard Delay after Transfer} = \frac{17}{\text{System Clock}}$$

Delay after transfer can be used to provide a peripheral deselect interval. A delay can also be inserted between consecutive transfers to allow serial A/D converters to complete conversion.

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[3:0]				0	0	0	0	NEWQP[3:0]			
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SPCR2 contains QSPI configuration parameters. The CPU can read and write this register; the QSM has read access only. Writes to SPCR2 are buffered. A write to SPCR2 that changes a bit value while the QSPI is operating is ineffective on the current serial transfer, but becomes effective on the next serial transfer. Reads of SPCR2 return the current value of the register, not of the buffer.

SPIFIE — SPI Finished Interrupt Enable

0 = QSPI interrupts disabled
1 = QSPI interrupts enabled

SPIFIE enables the QSPI to generate a CPU interrupt upon assertion of the status flag SPIF.

WREN — Wrap Enable

0 = Wraparound mode disabled
1 = Wraparound mode enabled

WREN enables or disables wraparound mode.

WRTO — Wrap To

When wraparound mode is enabled, after the end of queue has been reached, WRTO determines which address the QSPI executes.

Bit 12 — Not Implemented

ENDQP[3:0] — Ending Queue Pointer

This field contains the last QSPI queue address.

Bits [7:4] — Not Implemented

NEWQP[3:0] — New Queue Pointer Value

This field contains the first QSPI queue address.

SPCR3 — QSPI Control Register **$YFFC1E**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	LOOPQ	HMIE	HALT	SPSR							
RESET:															
0	0	0	0	0	0	0	0								

SPCR3 contains QSPI configuration parameters. The CPU can read and write SPCR3, but the QSM has read-only access.

Bits [15:11] — Not Implemented

LOOPQ — QSPI Loop Mode
0 = Feedback path disabled
1 = Feedback path enabled
LOOPQ controls feedback on the data serializer for testing.

HMIE — HALTA and MODF Interrupt Enable
0 = HALTA and MODF interrupts disabled
1 = HALTA and MODF interrupts enabled
HMIE controls CPU interrupts caused by the HALTA status flag or the MODF status flag in SPSR.

HALT — Halt
0 = Halt not enabled
1 = Halt enabled
When HALT is asserted, the QSPI stops on a queue boundary. It is in a defined state from which it can later be restarted.

SPSR — QSPI Status Register **$YFFC1F**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPCR3								SPIF	MODF	HALTA	0	CPTQP[3:0]			
RESET:															
								0	0	0	0	0	0	0	0

SPSR contains QSPI status information. Only the QSPI can assert the bits in this register. The CPU reads this register to obtain status information and writes it to clear status flags.

SPIF — QSPI Finished Flag
0 = QSPI not finished
1 = QSPI finished
SPIF is set after execution of the command at the address in ENDQP[3:0].

MODF — Mode Fault Flag
0 = Normal operation
1 = Another SPI node requested to become the network SPI master while the QSPI was enabled in master mode ($\overline{SS}$ input taken low).
The QSPI asserts MODF when the QSPI is the serial master (MSTR = 1) and the $\overline{SS}$ input pin is negated by an external driver.

HALTA — Halt Acknowledge Flag
0 = QSPI not halted
1 = QSPI halted
HALTA is asserted when the QSPI halts in response to CPU assertion of HALT.

Bit 4 — Not Implemented

CPTQP[3:0] — Completed Queue Pointer

CPTQP[3:0] points to the last command executed. It is updated when the current command is complete. When the first command in a queue is executing, CPTQP[3:0] contains either the reset value ($0) or a pointer to the last command completed in the previous queue.

5.5.3 QSPI RAM

The QSPI contains an 80-byte block of dual-access static RAM that is used by both the QSPI and the CPU. The RAM is divided into three segments: receive data, transmit data, and command control data. Receive data is information received from a serial device external to the MCU. Transmit data is information stored by the CPU for transmission to an external peripheral. Command control data is used to perform the transfer.

Figure 19 displays the organization of the RAM.

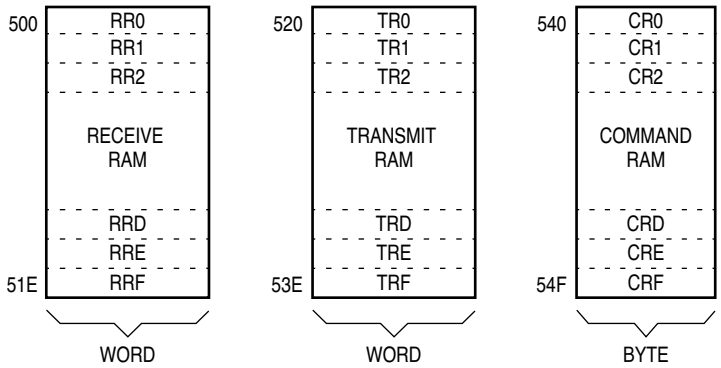


Figure 19 QSPI RAM

Once the CPU has set up the queue of QSPI commands and enabled the QSPI, the QSPI can operate independently of the CPU. The QSPI executes all of the commands in its queue, sets a flag indicating that it is finished, and then either interrupts the CPU or waits for CPU intervention. It is possible to execute a queue of commands repeatedly without CPU intervention.

RR[0:F] — Receive Data RAM **$YFFD00**

Data received by the QSPI is stored in this segment. The CPU reads this segment to retrieve data from the QSPI. Data stored in receive RAM is right-justified. Unused bits in a receive queue entry are set to zero by the QSPI upon completion of the individual queue entry. The CPU can access the data using byte, word, or long-word addressing.

The CPTQP[3:0] value in SPSR shows which queue entries have been executed. The CPU uses this information to determine which locations in receive RAM contain valid data before reading them.

TR[0:F] — Transmit Data RAM **$YFFD20**

Data that is to be transmitted by the QSPI is stored in this segment. The CPU usually writes one word of data into this segment for each queue command to be executed.

Information to be transmitted must be written to transmit data RAM in a right-justified format. The QSPI cannot modify information in the transmit data RAM. The QSPI copies the information to its data serializer for transmission. Information remains in transmit RAM until overwritten.

CR[0:F] — Command RAM **$YFFD40**

7	6	5	4	3	2	1	0
CONT	BITSE	DT	DSCK	PCS3	PCS2	PCS1	PCS0[1]
–	–	–	–	–	–	–	–
CONT	BITSE	DT	DSCK	PCS3	PCS2	PCS1	PCS0[1]
COMMAND CONTROL				PERIPHERAL CHIP SELECT			

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

Command RAM is used by the QSPI when in master mode. The CPU writes one byte of control information to this segment for each QSPI command to be executed. The QSPI cannot modify information in command RAM.

Command RAM consists of 16 bytes. Each byte is divided into two fields. The peripheral chip-select field enables peripherals for transfer. The command control field provides transfer options.

A maximum of 16 commands can be in the queue. Queue execution by the QSPI proceeds from the address in NEWQP[3:0] through the address in ENDQP[3:0]. (Both of these fields are in SPCR2).

CONT — Continue

0 = Control of chip selects returned to PORTQS after transfer is complete.
1 = Peripheral chip selects remain asserted after transfer is complete.

BITSE — Bits per Transfer Enable

0 = 8 bits
1 = Number of bits set in BITS[3:0] field of SPCR0

DT — Delay after Transfer

The QSPI provides a variable delay at the end of serial transfer to facilitate the interface with peripherals that have a latency requirement. The delay between transfers is determined by the SPCR1 DTL[6:0] field.

DSCK — PCS to SCK Delay

0 = PCS valid to SCK transition is one-half SCK.
1 = SPCR1 DSCKL[6:0] field specifies delay from PCS valid to SCK.

PCS[3:0] — Peripheral Chip Select

Use peripheral chip-select bits to select an external device for serial data transfer. More than one peripheral chip select can be activated at a time, and more than one peripheral chip can be connected to each PCS pin, provided that proper fanout is observed.

5.5.4 Operating Modes

The QSPI operates in either master or slave mode. Master mode is used when the MCU originates data transfers. Slave mode is used when an external device initiates serial transfers to the MCU through the QSPI. Switching between the modes is controlled by MSTR in SPCR0. Before entering either mode, appropriate QSM and QSPI registers must be properly initialized.

In master mode, the QSPI executes a queue of commands defined by control bits in each command RAM queue entry. Chip-select pins are activated, data is transmitted from transmit RAM and received into receive RAM.

In slave mode, operation proceeds in response to $\overline{SS}$ pin activation by an external bus master. Operation is similar to master mode, but no peripheral chip selects are generated, and the number of bits transferred is controlled in a different manner. When the QSPI is selected, it automatically executes the next queue transfer to exchange data with the external device correctly.

Although the QSPI inherently supports multi-master operation, no special arbitration mechanism is provided. A mode fault flag (MODF) indicates a request for SPI master arbitration. System software must provide arbitration. Note that unlike previous SPI systems, MSTR is not cleared by a mode fault being set, nor are the QSPI pin output drivers disabled. The QSPI and associated output drivers must be disabled by clearing SPE in SPCR1.

5.6 SCI Submodule

The SCI submodule is used to communicate with external devices through an asynchronous serial bus. The SCI is fully compatible with the SCI systems found on other Motorola MCUs, such as the M68HC11 and M68HC05 Families.

5.6.1 SCI Pins

There are two unidirectional pins associated with the SCI. The SCI controls the transmit data (TXD) pin when enabled, whereas the receive data (RXD) pin remains a dedicated input pin to the SCI. TXD is available as a general-purpose I/O pin when the SCI transmitter is disabled. When used for I/O, TXD can be configured either as input or output, as determined by QSM register DDRQS.

Table 57 shows SCI pins and their functions.

Table 57 SCI Pins

Pin Names	Mnemonics	Mode	Function
Receive Data	RXD	Receiver Disabled Receiver Enabled	Not Used Serial Data Input to SCI
Transmit Data	TXD	Transmitter Disabled Transmitter Enabled	General-Purpose I/O Serial Data Output from SCI

5.6.2 SCI Registers

The SCI programming model includes QSM global and pin control registers, and four SCI registers. There are two SCI control registers, one status register, and one data register. All registers can be read or written at any time by the CPU.

Changing the value of SCI control bits during a transfer operation may disrupt operation. Before changing register values, allow the transmitter to complete the current transfer, then disable the receiver and transmitter. Status flags in the SCSR may be cleared at any time.

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[12:0]												
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0

SCCR0 contains a baud rate selection parameter. Baud rate must be set before the SCI is enabled. The CPU can read and write this register at any time.

Bits [15:13] — Not Implemented

SCBR[12:0] — Baud Rate

SCI baud rate is programmed by writing a 13-bit value to BR. The baud rate is derived from the MCU system clock by a modulus counter.

The SCI receiver operates asynchronously. An internal clock is necessary to synchronize with an incoming data stream. The SCI baud rate generator produces a receiver sampling clock with a frequency 16 times that of the expected baud rate of the incoming data. The SCI determines the position of bit boundaries from transitions within the received waveform, and adjusts sampling points to the proper positions within the bit period. Receiver sampling rate is always 16 times the frequency of the SCI baud rate, which is calculated as follows:

$$\text{SCI Baud Rate} = \frac{\text{System Clock}}{32 \times \text{SCBR[12:0]}}$$

or

$$\text{SCBR[12:0]} = \frac{\text{System Clock}}{32 \times \text{SCI Baud Rate Desired}}$$

where SCBR[12:0] is in the range {1, 2, 3, …, 8191}

Writing a value of zero to SCBR[12:0] disables the baud rate generator.

Table 58 lists the SCBR[12:0] settings for standard and maximum baud rates using a 20.97 MHz system clock.

Table 58 SCI Baud Rates

Nominal Baud Rate	Actual Rate with 20.97 MHz Clock	SCBR[12:0] Value
64	—	—
110	110.0	$1745
300	300.1	$0888
600	600.1	$0444
1200	1200.3	$0222
2400	2400.6	$0111
4800	4783.6	$0089
9600	9637.6	$0044
19200	19275.3	$0022
38400	38550.6	$0011
76800	72817.8	$0009
Maximum Rate	655360.0	$0001

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

SCCR1 contains SCI configuration parameters. The CPU can read and write this register at any time. The SCI can modify RWU in some circumstances. In general, interrupts enabled by these control bits are cleared by reading SCSR, then reading (for receiver status bits) or writing (for transmitter status bits) SCDR.

Bit 15 — Not Implemented

LOOPS — Loop Mode

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

LOOPS controls a feedback path on the data serial shifter. When loop mode is enabled, SCI transmitter output is fed back into the receive serial shifter. TXD is asserted (idle line). Both transmitter and receiver must be enabled before entering loop mode.

WOMS — Wired-OR Mode for SCI Pins

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

WOMS determines whether the TXD pin is an open-drain output or a normal CMOS output. This bit is used only when TXD is an output. If TXD is used as a general-purpose input pin, WOMS has no effect.

ILT — Idle-Line Detect Type

0 = Short idle-line detect (start count on first one)
1 = Long idle-line detect (start count on first one after stop bit(s))

PT — Parity Type

0 = Even parity
1 = Odd parity

When parity is enabled, PT determines whether parity is even or odd for both the receiver and the transmitter.

PE — Parity Enable

0 = SCI parity disabled
1 = SCI parity enabled

PE determines whether parity is enabled or disabled for both the receiver and the transmitter. If the received parity bit is not correct, the SCI sets the PF error flag in SCSR.

When PE is set, the most significant bit (MSB) of the data field is used for the parity function, which results in either seven or eight bits of user data, depending on the condition of M bit. **Table 59** lists the available choices.

Table 59 Parity Enable Data Bit Selection

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

WAKE — Wakeup by Address Mark
0 = SCI receiver awakened by idle-line detection
1 = SCI receiver awakened by address mark (last bit set)

TIE — Transmit Interrupt Enable
0 = SCI TDRE interrupts inhibited
1 = SCI TDRE interrupts enabled

TCIE — Transmit Complete Interrupt Enable
0 = SCI TC interrupts inhibited
1 = SCI TC interrupts enabled

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

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

TE — Transmitter Enable
0 = SCI transmitter disabled (TXD pin may be used as I/O)
1 = SCI transmitter enabled (TXD pin dedicated to SCI transmitter)
The transmitter retains control of the TXD pin until completion of any character transfer that was in progress when TE is cleared.

RE — Receiver Enable
0 = SCI receiver disabled (status bits inhibited)
1 = SCI receiver enabled

RWU — Receiver Wakeup
0 = Normal receiver operation (received data recognized)
1 = Wakeup mode enabled (received data ignored until awakened)
Setting RWU enables the wakeup function, which allows the SCI to ignore received data until awakened by either an idle line or address mark (as determined by WAKE). When in wakeup mode, the receiver status flags are not set, and interrupts are inhibited. This bit is cleared automatically (returned to normal mode) when the receiver is awakened.

SBK — Send Break
0 = Normal operation
1 = Break frame(s) transmitted after completion of current frame
SBK provides the ability to transmit a break code from the SCI. If the SCI is transmitting when SBK is set, it will transmit continuous frames of zeros after it completes the current frame, until SBK is cleared. If SBK is toggled (one to zero in less than one frame interval), the transmitter sends only one or two break frames before reverting to idle line or beginning to send data.

SCSR — SCI Status Register **$YFFC0C**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NOT USED							TDRE	TC	RDRF	RAF	IDLE	OR	NF	FE	PF
RESET:							1	1	0	0	0	0	0	0	0

SCSR contains flags that show SCI operational conditions. These flags can be cleared either by hardware or by a special acknowledgment sequence. The sequence consists of SCSR read with flags set, followed by SCDR read (write in the case of TDRE and TC). A long-word read can consecutively access both SCSR and SCDR. This action clears receive status flag bits that were set at the time of the read, but does not clear TDRE or TC flags.

If an internal SCI signal for setting a status bit comes after the CPU has read the asserted status bits, but before the CPU has written or read register SCDR, the newly set status bit is not cleared. SCSR must be read again with the bit set. Also, SCDR must be written or read before the status bit is cleared.

Reading either byte of SCSR causes all 16 bits to be accessed. Any status bit already set in either byte will be cleared on a subsequent read or write of register SCDR.

TDRE — Transmit Data Register Empty Flag

0 = Register TDR still contains data to be sent to the transmit serial shifter.
1 = A new character can now be written to the transmit data register.

TDRE is set when the byte in the transmit data register is transferred to the transmit serial shifter. If TDRE is zero, transfer has not occurred and a write to the transmit data register will overwrite the previous value. New data is not transmitted if the transmit data register is written without first clearing TDRE.

TC — Transmit Complete Flag

0 = SCI transmitter is busy
1 = SCI transmitter is idle

TC is set when the transmitter finishes shifting out all data, queued preambles (mark/idle line), or queued breaks (logic zero). The interrupt can be cleared by reading SCSR when TC is set and then by writing the transmit data register of SCDR.

RDRF — Receive Data Register Full Flag

0 = Receive data register is empty or contains previously read data.
1 = Receive data register contains new data.

RDRF is set when the content of the receive serial shifter is transferred to the receive data register. If one or more errors are detected in the received word, flag(s) NF, FE, and/or PF are set within the same clock cycle.

RAF — Receiver Active Flag

0 = SCI receiver is idle
1 = SCI receiver is busy

RAF indicates whether the SCI receiver is busy. It is set when the receiver detects a possible start bit and is cleared when the chosen type of idle line is detected. RAF can be used to reduce collisions in systems with multiple masters.

IDLE — Idle-Line Detected Flag

0 = SCI receiver did not detect an idle-line condition.
1 = SCI receiver detected an idle-line condition.

IDLE is disabled when RWU in SCCR1 is set. IDLE is set when the SCI receiver detects the idle-line condition specified by ILT in SCCR1. If cleared, IDLE will not set again until after RDRF is set. RDRF is set when a break is received, so that a subsequent idle line can be detected.

OR — Overrun Error Flag

0 = RDRF is cleared before new data arrives.
1 = RDRF is not cleared before new data arrives.

OR is set when a new byte is ready to be transferred from the receive serial shifter to the receive data register, and RDRF is still set. Data transfer is inhibited until OR is cleared. Previous data in receive data register remains valid, but data received during overrun condition (including the byte that set OR) is lost.

NF — Noise Error Flag

0 = No noise detected on the received data

1 = Noise occurred on the received data

NF is set when the SCI receiver detects noise on a valid start bit, on any data bit, or on a stop bit. It is not set by noise on the idle line or on invalid start bits. Each bit is sampled three times. If none of the three samples are the same logic level, the majority value is used for the received data value, and NF is set. NF is not set until an entire frame is received and RDRF is set.

FE — Framing Error Flag

0 = No framing error on the received data.

1 = Framing error or break occurred on the received data.

FE is set when the SCI receiver detects a zero where a stop bit was to have occurred. FE is not set until the entire frame is received and RDRF is set. A break can also cause FE to be set. It is possible to miss a framing error if RXD happens to be at logic level one at the time the stop bit is expected.

PF — Parity Error Flag

0 = No parity error on the received data

1 = Parity error occurred on the received data

PF is set when the SCI receiver detects a parity error. PF is not set until the entire frame is received and RDRF is set.

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

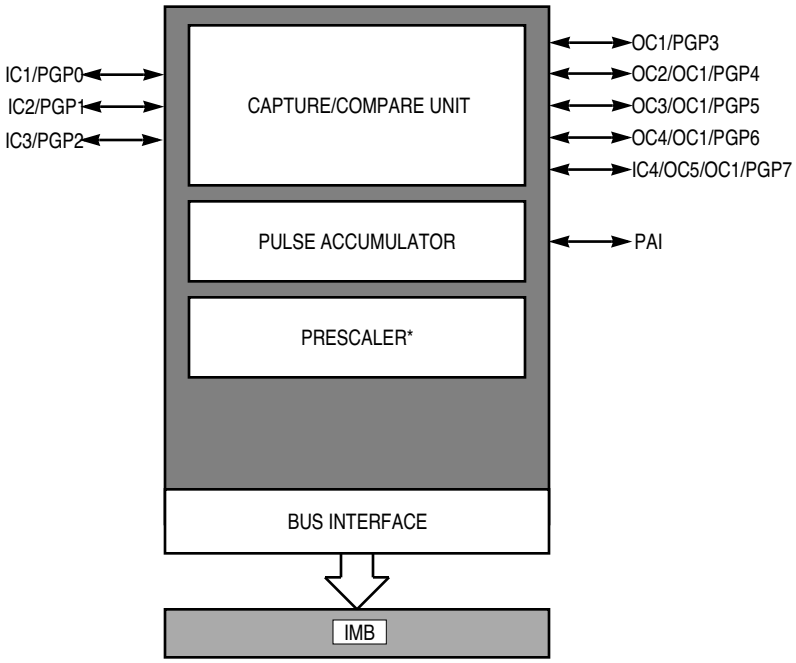
SCDR contains two data registers at the same address. The receive data register is a read-only register that contains data received by the SCI. The data comes into the receive serial shifter and is transferred to the receive data register. The transmit data register is a write-only register that contains data to be transmitted. The data is first written to the transmit data register, then transferred to the transmit serial shifter, where additional format bits are added before transmission. R[7:0]/T[7:0] contain either the first eight data bits received when SCDR is read, or the first eight data bits to be transmitted when SCDR is written. R8/T8 are used when the SCI is configured for 9-bit operation. When it is configured for 8-bit operation, they have no meaning or effect.

6 General-Purpose Timer Module

The GPT is a simple, yet flexible 9-channel timer used in systems where a moderate degree of external visibility and control is required. The GPT consists of a capture/compare unit, a pulse accumulator, and a prescaler. A bus interface unit connects the GPT to the intermodule bus. Figure 20 shows a block diagram of the GPT.

NOTE

On the MC68HC16V1, the external clock input signal (PCLK) is tied so that a logical zero is indicated as the pin state under any condition.



* SOME VERSIONS OF THE GPT CAN USE AN EXTERNAL CLOCK SOURCE. HOWEVER, THE GPT EXTERNAL CLOCK INPUT SIGNAL (PCLK) IN THIS MCU IS NOT BONDED TO AN EXTERNAL PIN.

V1 GPT BLOCK

Figure 20 GPT Block Diagram

6.1 Overview

The capture/compare unit features three input capture channels, four output compare channels, and one selectable input capture/output compare channel. These channels share a 16-bit free-running counter (TCNT) which derives its clock from a nine-state prescaler.

Pulse accumulator channel logic includes an 8-bit counter. The pulse accumulator can operate in either event counting mode or gated time accumulation mode.

All GPT pins can also be used for general-purpose input/output. The input capture and output compare pins form a bidirectional 8-bit parallel port (PORTGP).

GPT input capture/output compare pins are bidirectional and can be used to form an 8-bit parallel port.

6.2 Address Map

Table 60 shows the GPT address map.

Table 60 GPT Address Map

Address	15 8	7 0
$YFF900[1]	GPT MODULE CONFIGURATION REGISTER (GPTMCR)	
$YFF902	GPT MODULE TEST REGISTER (GPTMTR)	
$YFF904	INTERRUPT CONFIGURATION REGISTER (ICR)	
$YFF906	PGP DATA DIRECTION (DDRGP)	PGP DATA (PORTGP)
$YFF908	OC1 ACTION MASK (OC1M)	OC1 ACTION DATA (OC1D)
$YFF90A	TIMER COUNTER (TCNT)	
$YFF90C	PULSE ACCUMULATOR CONTROL (PACTL)	PULSE ACCUMULATOR COUNTER (PACNT)
$YFF90E	TIMER INPUT CAPTURE 1 (TIC1)	
$YFF910	TIMER INPUT CAPTURE 2 (TIC2)	
$YFF912	TIMER INPUT CAPTURE 3 (TIC3)	
$YFF914	TIMER OUTPUT COMPARE 1 (TOC1)	
$YFF916	TIMER OUTPUT CAPTURE 2 (TOC2)	
$YFF918	TIMER OUTPUT CAPTURE 3 (TOC3)	
$YFF91A	TIMER OUTPUT CAPTURE 4 (TOC4)	
$YFF91C	TIMER INPUT CAPTURE 4/OUTPUT COMPARE 5 (TI4/O5)	
$YFF91E	TIMER CONTROL 1 (TCTL1)	TIMER CONTROL 2 (TCTL2)
$YFF920	TIMER MASK 1 (TMSK1)	TIMER MASK 2 (TMSK2)
$YFF922	TIMER FLAG 1 (TFLG1)	TIMER FLAG 2 (TFLG2)
$YFF924	FORCE COMPARE (CFORC)	NOT USED
$YFF926	NOT USED	
$YFF928	NOT USED	
$YFF92A	NOT USED	
$YFF92C	GPT PRESCALER (PRESCL)	
$YFF92E–$YFF93F	RESERVED	

1. Y = M111, where M is the logic state of the module mapping (MM) bit in the SLIMCR.

6.3 Capture/Compare Unit

The capture/compare unit features three input capture channels, four output compare channels, and one input capture/output compare channel (function selected by control register). Refer to **Figure 21**.

These channels share a 16-bit free-running counter (TCNT), which derives its clock from seven stages of a 9-stage prescaler. Refer to **Figure 22**.

This section also contains one pulse accumulator channel. The pulse accumulator logic includes its own 8-bit counter and can operate in either event counting mode or gated time accumulation mode. Refer to **Figure 23**.

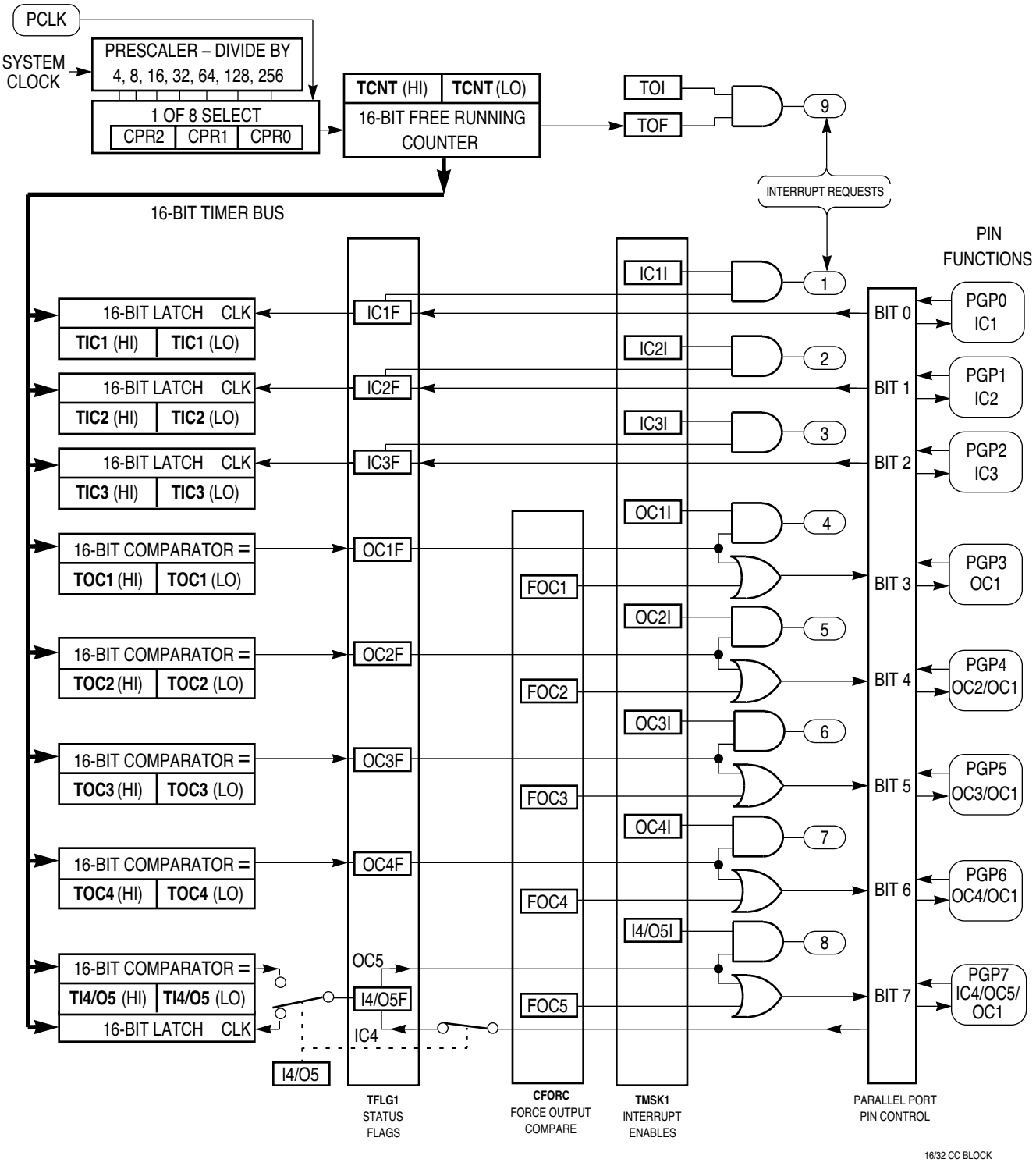


Figure 21 GPT Capture/Compare Block Diagram

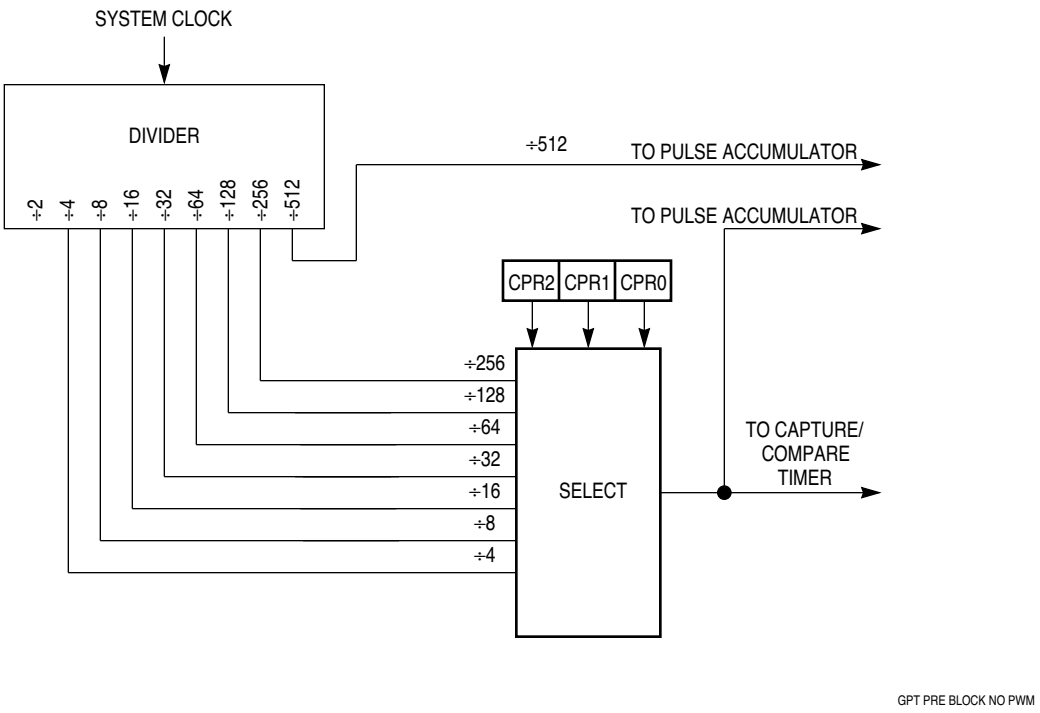


Figure 22 Prescaler Block Diagram

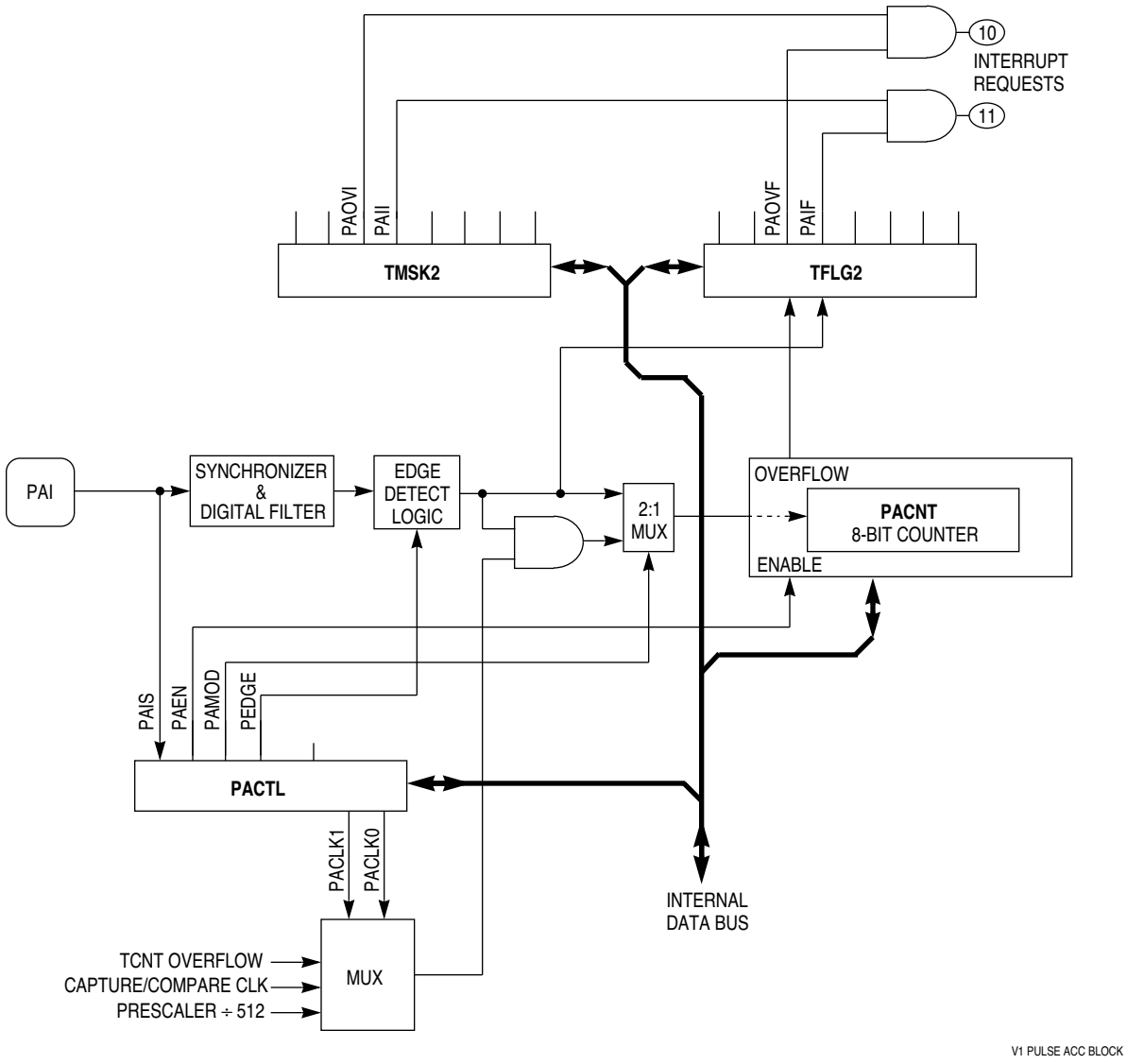


Figure 23 Pulse Accumulator Block Diagram

6.4 GPT Registers

The following section provides a summary of GPT registers and their contents.

GPTMCR — GPT Module Configuration Register **$YFF900**

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

The GPTMCR contains parameters for configuring the GPT.

STOP — Stop Clocks
- 0 = Internal clocks not shut down
- 1 = Internal clocks shut down

FRZ1 — Not Implemented

FRZ0 — FREEZE Assertion Response
- 0 = Ignore IMB FREEZE Signal
- 1 = FREEZE the current state of the GPT

STOPP — Stop Prescaler
- 0 = Normal operation
- 1 = Stop prescaler and pulse accumulator from incrementing. Ignore changes to input pins.

INCP — Increment Prescaler
- 0 = Has no meaning
- 1 = If STOPP is asserted, increment prescaler once and clock input synchronizers once.

SUPV — Supervisor/Unrestricted Data Space
This bit has no effect because the CPU16 always operates in the supervisor mode.

IARB[3:0] — Interrupt Arbitration Identification
The value in this field is used to arbitrate between simultaneous interrupt service requests of the same priority. Each module that can generate interrupts has an IARB field. In order to implement an arbitration scheme, each IARB field must be set to a different non-zero value. If an interrupt request from a module that has an IARB field value of $0 is recognized, the CPU16 processes a spurious interrupt exception. The reset value of all IARB fields other than that of the SCIM is $0 (no priority), to preclude interrupt processing during reset.

GPTMTR — GPT Module Test Register **$YFF902**

This address is reserved for GPT factory test.

ICR — GPT Interrupt Configuration Register **$YFF904**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IPA[3:0]				0	IPL[2:0]			IVBA[3:0]				0	0	0	0
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPA[3:0] — Interrupt Priority Adjust
This field specifies which GPT interrupt source is given highest internal priority. Refer to **Table 61**.

Table 61 GPT Interrupt Sources

Name	Source Number	Source	Vector Number
—	0000	Adjusted Channel	IVBA : 0000
IC1	0001	Input Capture 1	IVBA : 0001
IC2	0010	Input Capture 2	IVBA : 0010
IC3	0011	Input Capture 3	IVBA : 0011
OC1	0100	Output Compare 1	IVBA : 0100
OC2	0101	Output Compare 2	IVBA : 0101
OC3	0110	Output Compare 3	IVBA : 0110
OC4	0111	Output Compare 4	IVBA : 0111
IC4/OC5	1000	Input Capture 4/Output Compare 5	IVBA : 1000
TO	1001	Timer Overflow	IVBA : 1001
PAOV	1010	Pulse Accumulator Overflow	IVBA : 1010
PAI	1011	Pulse Accumulator Input	IVBA : 1011

IPL[2:0] — Interrupt Priority Level
This field specifies the priority level of interrupts generated by the GPT.

IVBA[3:0] — Interrupt Vector Base Address
Most significant nibble of interrupt vector numbers generated by the GPT. Refer to **Table 61**.

DDRGP/PORTGP — Port GP Data Direction Register/Port GP Data Register **$YFF906**

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

When GPT pins are used as an 8-bit port, DDRGP determines whether pins are input or output and PORTGP holds the 8-bit data.

DDRGP[7:0] — Port GP Data Direction Register
0 = Input only
1 = Output

OC1M/OC1D — OC1 Action Mask Register/OC1 Action Data Register **$YFF908**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OC1M[5:1]					0	0	0	OC1D[5:1]					0	0	0
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

All OC outputs can be controlled by the action of OC1. OC1M contains a mask that determines which pins are affected. OC1D determines what the outputs are.

OC1M[5:1] — OC1 Mask Field
OC1M[5:1] correspond to OC[5:1].
0 = Corresponding output compare pin is not affected by OC1 compare.
1 = Corresponding output compare pin is affected by OC1 compare.

OC1D[5:1] — OC1 Data Field

OC1D[5:1] correspond to OC[5:1].

0 = If OC1 mask bit is set, clear the corresponding output compare pin on OC1 match.
1 = If OC1 mask bit is set, set the corresponding output compare pin on OC1 match.

TCNT — Timer Counter Register **$YFF90A**

TCNT is the 16-bit free-running counter associated with the input capture, output compare, and pulse accumulator functions of the GPT module.

PACTL/PACNT — Pulse Accumulator Control Register/Counter **$YFF90C**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PAIS	PAEN	PAMOD	PEDGE	PCLKS	I4/O5	PACLK[1:0]		PULSE ACCUMULATOR COUNTER							
RESET:															
U	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PACTL enables the pulse accumulator and selects either event counting or gated mode. In event counting mode, PACNT is incremented each time an event occurs. In gated mode, it is incremented by an internal clock.

PAIS — PAI Pin State (Read Only)

PAEN — Pulse Accumulator Enable

0 = Pulse accumulator disabled
1 = Pulse accumulator enabled

PAMOD — Pulse Accumulator Mode

0 = External event counting
1 = Gated time accumulation

PEDGE — Pulse Accumulator Edge Control

The effects of PAMOD and PEDGE are shown in **Table 62**.

Table 62 PAMOD/PEDGE Effects

PAMOD	PEDGE	Effect
0	0	PAI falling edge increments counter
0	1	PAI rising edge increments counter
1	0	Zero on PAI inhibits counting
1	1	One on PAI inhibits counting

I4/O5 — Input Capture 4/Output Compare 5

0 = Output compare 5 enabled
1 = Input capture 4 enabled

PACLK[1:0] — Pulse Accumulator Clock Select (Gated Mode)

Table 63 shows the PACLK[1:0] bit field effects.

Table 63 PACLK[1:0] Bit Field

PACLK[1:0]	Pulse Accumulator Clock Selected
00	System clock divided by 512
01	Same clock used to increment TCNT
10	TOF flag from TCNT
11	External clock, PCLK

PACNT — Pulse Accumulator Counter

Eight-bit read/write counter used for external event counting or gated time accumulation.

TIC[1:3] — Input Capture Registers 1–3 **$YFF90E, $YFF910, $YFF912**

The input capture registers are 16-bit read-only registers which are used to latch the value of TCNT when a specified transition is detected on the corresponding input capture pin. They are reset to $FFFF.

TOC[1:4] — Output Compare Registers 1–4 **$YFF914, $YFF916, $YFF918, $YFF91A**

The output compare registers are 16-bit read/write registers which can be used as output waveform controls or as elapsed time indicators. For output compare functions, they are written to a desired match value and compared against TCNT to control specified pin actions. They are reset to $FFFF.

TI4/O5 — Input Capture 4/Output Compare 5 Register **$YFF91C**

This register serves either as input capture register 4 or output compare register 5, depending on the state of I4/O5 in PACTL. It is reset to $FFFF.

TCTL1/TCTL2 — Timer Control Registers 1–2 **$YFF91E**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OM5	OL5	OM4	OL4	OM3	OL3	OM2	OL2	EDG4B	EDG4A	EDG3B	EDG3A	EDG2B	EDG2A	EDG1B	EDG1A
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TCTL1 determines output compare mode and output logic level. TCTL2 determines the type of input capture to be performed.

OM/OL[5:2] — Output Compare Mode Bits and Output Compare Level Bits

Each pair of bits specifies an action to be taken when output comparison is successful. Refer to **Table 64**.

Table 64 OM/OL[5:2] Bit Field Effects

OM/OL[5:2]	Action Taken
00	Timer disconnected from output logic
01	Toggle OCx output line
10	Clear OCx output line to zero
11	Set OCx output line to one

EDG[4:1]B/A — Input Capture Edge Control Bits

Each pair of bits configures input sensing logic for the corresponding input capture. Refer to **Table 65**.

Table 65 EDG[4:1]B/A Bit Field Effects

EDG[4:1]B/A	Configuration
00	Capture disabled
01	Capture on rising edge only
10	Capture on falling edge only
11	Capture on any (rising or falling) edge

TMSK1/TMSK2 — Timer Interrupt Mask Registers 1–2 **$YFF920**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I4/O5I	OCI[4:1]				ICI[3:1]			TOI	0	PAOVI	PAII	CPROUT	CPR[2:0]		
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TMSK1 enables OC and IC interrupts. TMSK2 controls pulse accumulator interrupts and TCNT functions.

I4/O5I — Input Capture 4/Output Compare 5 Interrupt Enable
0 = IC4/OC5 interrupt disabled.
1 = IC4/OC5 interrupt requested when I4/O5F in TFLG1 is set.

OCI[4:1] — Output Compare Interrupt Enable
OCI[4:1] correspond to OC[4:1].
0 = OC interrupt disabled
1 = OC interrupt requested when OC flag set

ICI[3:1] — Input Capture Interrupt Enable
ICI[3:1] correspond to IC[3:1].
0 = IC interrupt disabled
1 = IC interrupt requested when IC flag set

TOI — Timer Overflow Interrupt Enable
0 = Timer overflow interrupt disabled
1 = Interrupt requested when TOF is set

PAOVI — Pulse Accumulator Overflow Interrupt Enable
0 = Pulse accumulator overflow interrupt disabled
1 = Interrupt requested when PAOVF is set

PAII — Pulse Accumulator Input Interrupt Enable
0 = Pulse accumulator interrupt disabled
1 = Interrupt requested when PAIF is set

CPROUT — Compare/Capture Unit Clock Output Enable
0 = Normal operation for OC1 pin
1 = TCNT clock driven out OC1 pin

CPR[2:0] — Timer Prescaler/PCLK Select Field
This field selects one of seven prescaler taps or PCLK to be TCNT input. Refer to **Table 66**.

Table 66 CPR[2:0] Bit Field Effects

CPR[2:0]	System Clock Divide-by Factor
000	4
001	8
010	16
011	32
100	64
101	128
110	256
111	PCLK

TFLG1/TFLG2 — Timer Interrupt Flag Registers 1–2 **$YFF922**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I4/O5F	OCF[4:1]				ICF[3:1]			TOF	0	PAOVF	PAIF	0	0	0	0
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

These registers show condition flags that correspond to various GPT events. If the corresponding interrupt enable bit in TMSK1/TMSK2 is set, an interrupt occurs.

I4/O5F — Input Capture 4/Output Compare 5 Flag
When I4/O5 in PACTL is zero, this flag is set each time TCNT matches the value in TOC5. When I4/O5 in PACTL is one, the flag is set each time a selected edge is detected at the I4/O5 pin.

OCF[4:1] — Output Compare Flags
An output compare flag is set each time TCNT matches the corresponding TOC register. OCF[4:1] correspond to OC[4:1].

ICF[3:1] — Input Capture Flags
A flag is set each time a selected edge is detected at the corresponding input capture pin. ICF[3:1] correspond to IC[3:1].

TOF — Timer Overflow Flag
This flag is set each time TCNT advances from a value of $FFFF to $0000.

PAOVF — Pulse Accumulator Overflow Flag
This flag is set each time the pulse accumulator counter advances from a value of $FF to $00.

PAIF — Pulse Accumulator Flag
In event counting mode, this flag is set when an active edge is detected on the PAI pin. In gated time accumulation mode, PAIF is set at the end of the timed period.

CFORC — Compare Force Register/PWM Control Register **$YFF924**

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

Setting a bit in CFORC causes a specific output on OC.

FOC[5:1] — Force Output Compare
FOC[5:1] correspond to OC[5:1].
0 = Causes no action on corresponding output compare pin.
1 = Causes pin action programmed for corresponding OC pin, but the OC flag is not set.

PRESCL — GPT Prescaler **$YFF92C**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UNUSED															
POWER ON RESET ONLY:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The 9-bit prescaler value can be read from bits [8:0] at this address. Bits [15:9] always read as zeros. Reset state is $0000.

NOTES

NOTES

Home Page:
www.freescale.com
email:
support@freescale.com
USA/Europe or Locations Not Listed:
Freescale Semiconductor
Technical Information Center, CH370
1300 N. Alma School Road
Chandler, Arizona 85224
(800) 521-6274
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
+81 2666 8080
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
(800) 441-2447
303-675-2140
Fax: 303-675-2150
LDCForFreescaleSemiconductor
@hibbertgroup.com

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

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