MOTOROLA
SEMICONDUCTOR
TECHNICAL DATA

Order this document
by MC68HC16Z1TS/D Rev. 3

MC68HC16Z1

Technical Summary
16-Bit Microcontroller

1 Introduction

The MC68HC16Z1 is a high-speed 16-bit control unit that is upwardly code compatible with M68HC11 controllers. 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 MC68HC16Z1 incorporates a true 16-bit central processing unit (CPU16), a system integration module (SIM), an 8/10-bit analog-to-digital converter (ADC), a queued serial module (QSM), a general-purpose timer (GPT), and a 2048-byte standby RAM (SRAM). These modules are interconnected by the intermodule bus (IMB).

Maximum system clock for the MC68HC16Z1 is 16.78 MHz. A phase-locked loop circuit synthesizes the clock from a frequency reference. Either a crystal (nominal frequency: 32.768 kHz) or an externally generated signal can be used. System hardware and software support changes in clock rate during operation. Because the MC68HC16Z1 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 MC68HC16Z1 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.

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

© MOTOROLA INC., 1992, 1996

Table 1 Ordering Information

Device Package	Temperature Range (°C)	Reference Frequency	Shipping Method	Order Number
132-PIN PLASTIC SURFACE MOUNT	–40 to 85	16.78 MHz	36 PER TRAY	XC16Z1CFC16
			2 PER TRAY	SPAKXC16Z1CFC16
		20 MHz	36 PER TRAY	XC16Z1CFC20
			2 PER TRAY	SPAKXC16Z1CFC20
		25 MHz	36 PER TRAY	XC16Z1CFC25
			2 PER TRAY	SPAKXC16Z1CFC25
	–40 to 105	16.78 MHz	36 PER TRAY	XC16Z1VFC16
			2 PER TRAY	SPAKXC16Z1VFC16
		20 MHz	36 PER TRAY	XC16Z1VFC20
			2 PER TRAY	SPAKXC16Z1VFC20
		25 MHz	36 PER TRAY	XC16Z1VFC25
			2 PER TRAY	SPAKXC16Z1VFC25
	–40 to 125	16.78 MHz	36 PER TRAY	XC16Z1MFC16
			2 PER TRAY	SPAKXC16Z1MFC16
		20 MHz	36 PER TRAY	XC16Z1MFC20
			2 PER TRAY	SPAKXC16Z1MFC20
		25 MHz	36 PER TRAY	XC16Z1MFC25
			2 PER TRAY	SPAKXC16Z1MFC25
132-PIN MOLDED CARRIER RING	–40 to 85	16.78 MHz	10 PER TUBE	XC16Z1CFD16
		20 MHz	10 PER TUBE	XC16Z1CFD20
		25 MHz	10 PER TUBE	XC16Z1CFD25
	–40 to 105	16.78 MHz	10 PER TUBE	XC16Z1VFD16
		20 MHz	10 PER TUBE	XC16Z1VFD20
		25 MHz	10 PER TUBE	XC16Z1VFD25
	–40 to 125	16.78 MHz	10 PER TUBE	XC16Z1MFD16
		20 MHz	10 PER TUBE	XC16Z1MFD20
		25 MHz	10 PER TUBE	XC16Z1MFD25
144-PIN PLASTIC SURFACE MOUNT	–40 to 85	16.78 MHz	44 PER TRAY	XC16Z1CFV16
			2 PER TRAY	SPAKXC16Z1CFV16
		20 MHz	44 PER TRAY	XC16Z1CFV20
			2 PER TRAY	SPAKXC16Z1CFV20
		25 MHz	44 PER TRAY	XC16Z1CFV25
			2 PER TRAY	SPAKXC16Z1CFV25
	–40 to 105	16.78 MHz	44 PER TRAY	XC16Z1VFV16
			2 PER TRAY	SPAKXC16Z1VFV16
		20 MHz	44 PER TRAY	XC16Z1VFV20
			2 PER TRAY	SPAKXC16Z1VFV20
		25 MHz	44 PER TRAY	XC16Z1VFV25
			2 PER TRAY	SPAKXC16Z1VFV25
	–40 to 125	16.78 MHz	44 PER TRAY	XC16Z1MFV16
			2 PER TRAY	SPAKXC16Z1MFV16
		20 MHz	44 PER TRAY	XC16Z1MFV20
			2 PER TRAY	SPAKXC16Z1MFV20
		25 MHz	44 PER TRAY	XC16Z1MFV25
			2 PER TRAY	SPAKXC16Z1MFV25
144-PIN MOLDED CARRIER RING	–40 to 85	16.78 MHz	13 PER TUBE	XC16Z1CFM16
		20 MHz	13 PER TUBE	XC16Z1CFM20
		25 MHz	13 PER TUBE	XC16Z1CFM25
	–40 to 105	16.78 MHz	13 PER TUBE	XC16Z1VFM16
		20 MHz	13 PER TUBE	XC16Z1VFM20
		25 MHz	13 PER TUBE	XC16Z1VFM25
	–40 to 125	16.78 MHz	13 PER TUBE	XC16Z1MFM16
		20 MHz	13 PER TUBE	XC16Z1MFM20
		25 MHz	13 PER TUBE	XC16Z1MFM25

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
  - Control-Oriented Digital Signal Processing Capability
  - 1 Megabyte of Program Memory and 1 Megabyte of Data Memory
  - High-Level Language Support
  - Fast Interrupt Response Time
  - Background Debugging Mode
  - Fully Static Operation
- System Integration Module
  - External Bus Support
  - Programmable Chip-Select Outputs
  - System Protection Logic
  - Watchdog Timer, Clock Monitor, and Bus Monitor
  - Two 8-Bit Dual Function Ports
  - One 7-Bit Dual Function Port
  - Phase-Locked Loop (PLL) Clock System
- 8/10-Bit Analog-to-Digital Converter
  - Eight Channels, Eight Result Registers
  - Eight Automated Modes
  - Three Result Alignment Modes
  - One 8-Bit Digital Input Port
- Queued Serial Module
  - Enhanced Serial Communication Interface
  - Queued Serial Peripheral Interface
  - One 8-Bit Dual Function Port
- General-Purpose Timer
  - Two 16-Bit Free-Running Counters with Prescaler
  - Three Input Capture Channels
  - Four Output Compare Channels
  - One Input Capture/Output Compare Channel
  - One Pulse Accumulator/Event Counter Input
  - Two Pulse Width Modulation Outputs
  - One 8-Bit Dual Function Port
  - Two Optional Discrete Inputs
  - Optional External Clock Input
- Standby RAM
  - 1024-Byte Static RAM
  - External Standby Voltage Supply Input

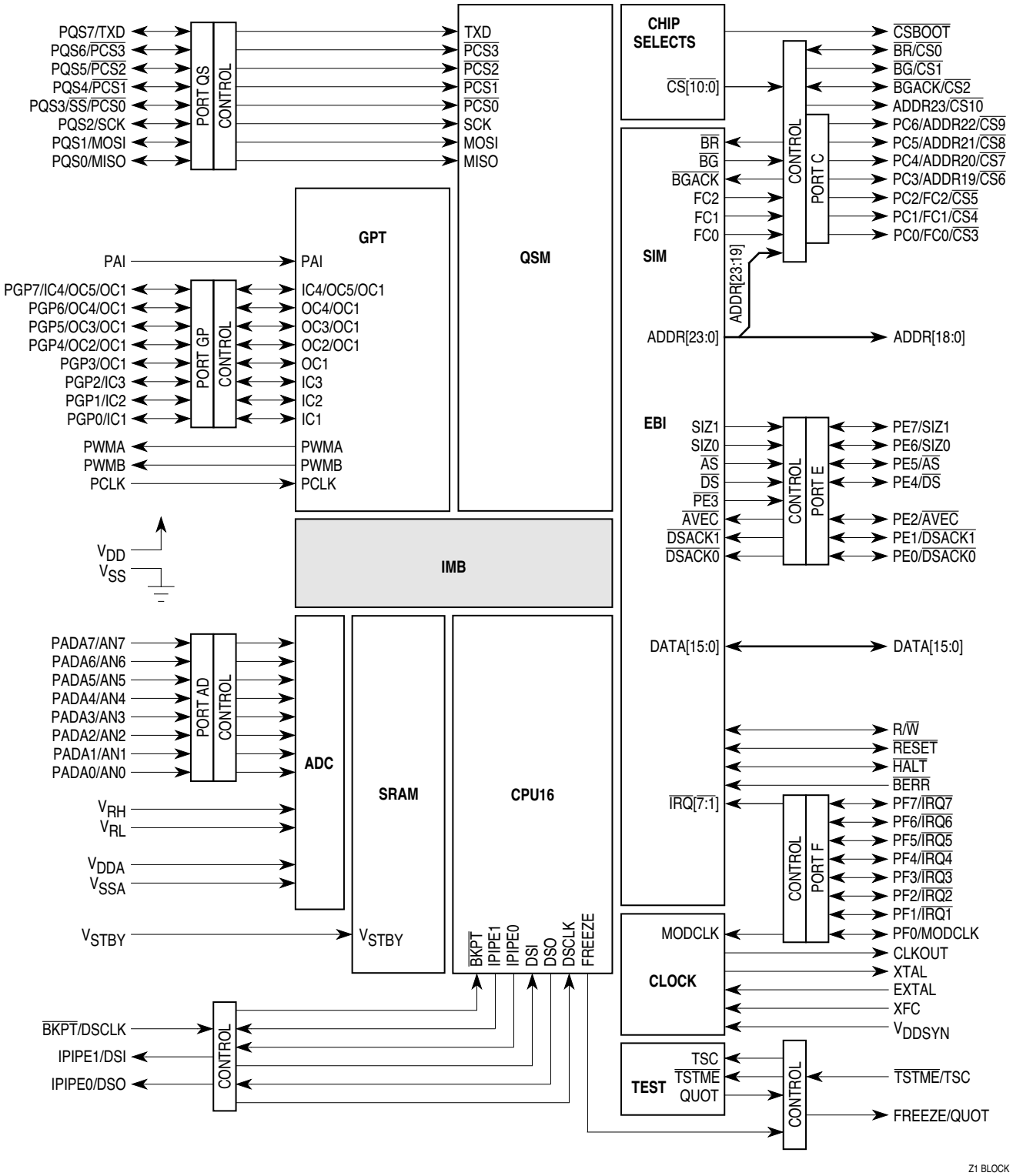


Figure 1 MC68HC16Z1 Block Diagram

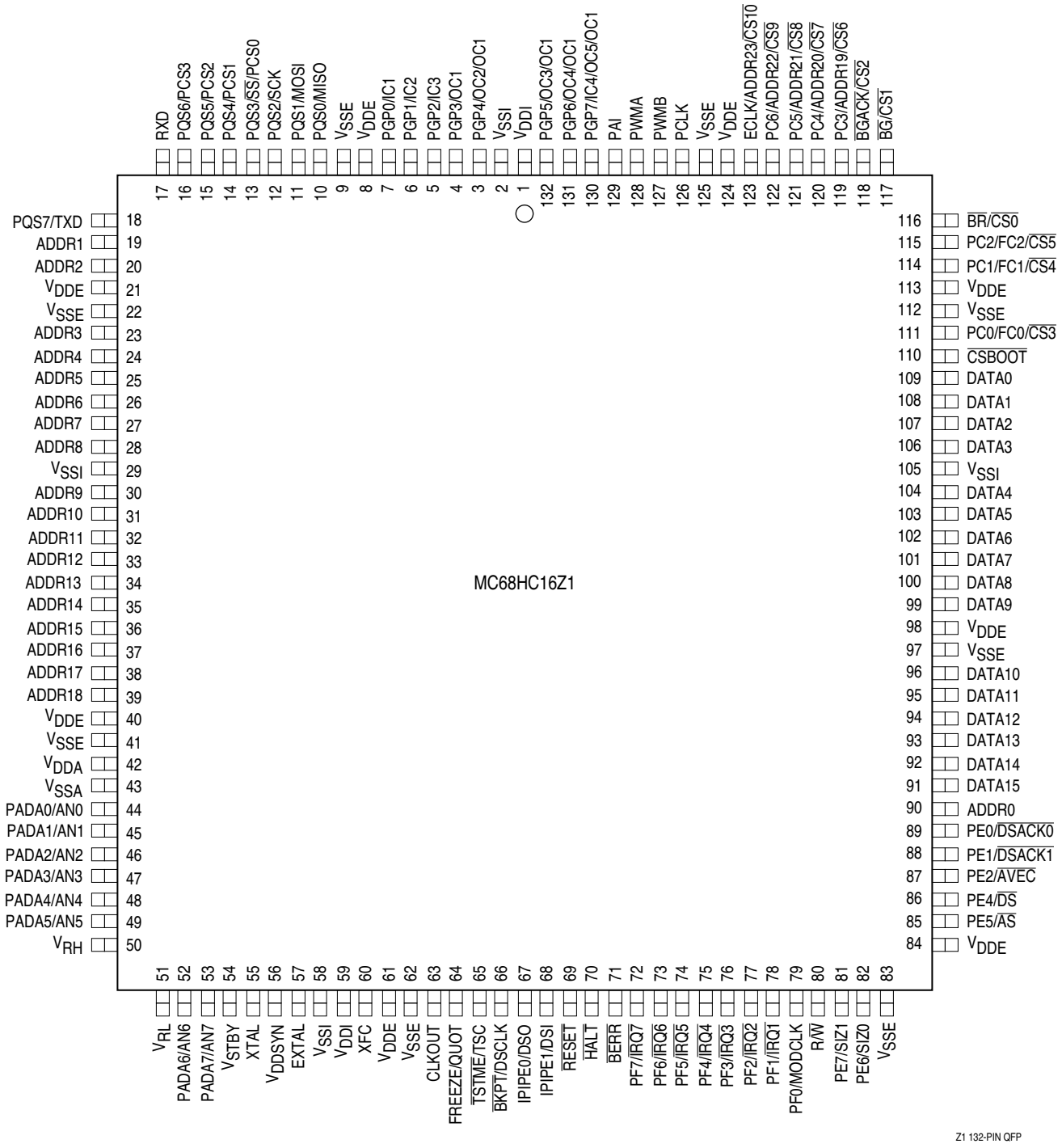


Figure 2 MC68HC16Z1 132-Pin Package Pin Assignments

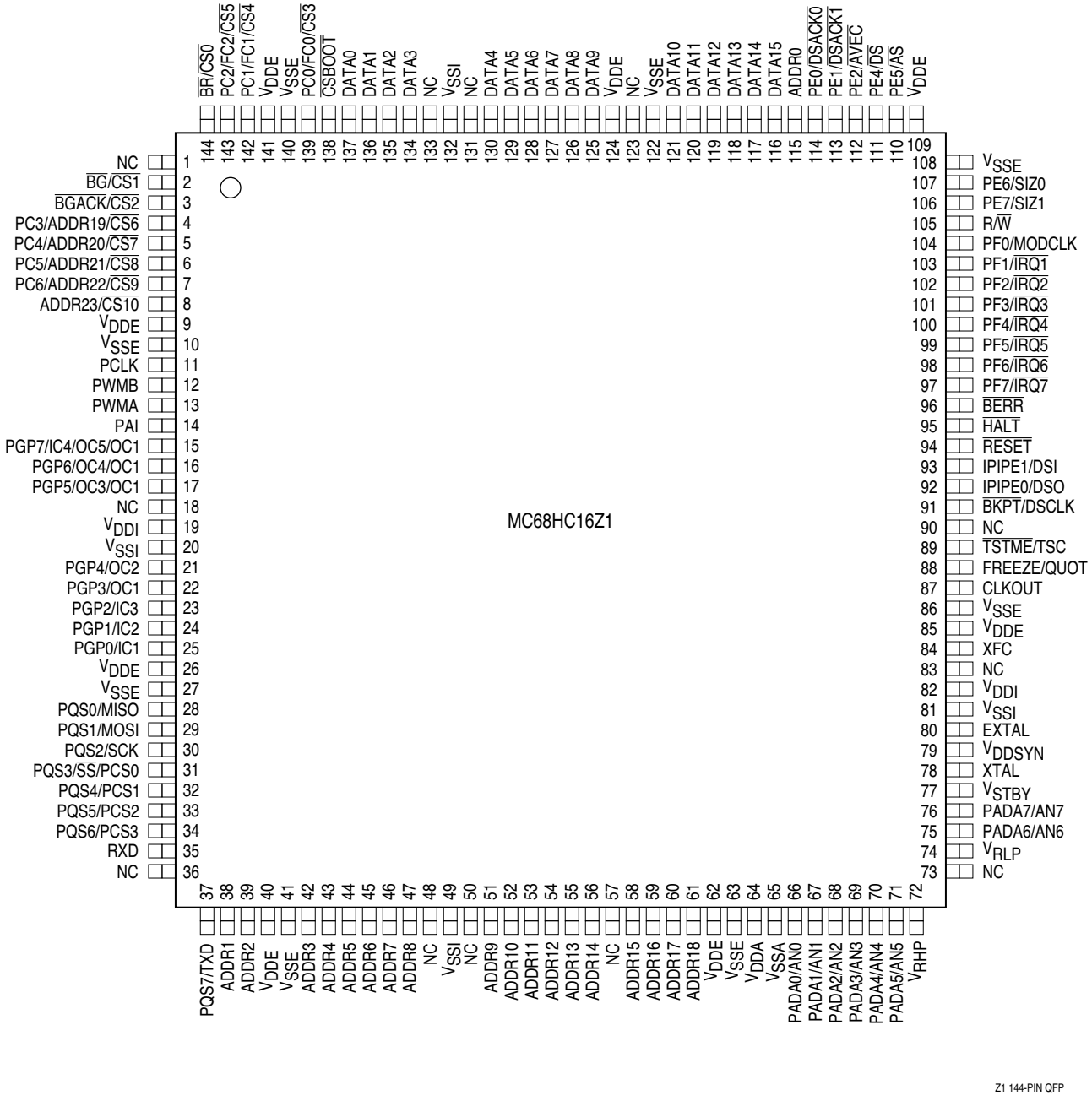


Figure 3 MC68HC16Z1 144-Pin Package Pin Assignments

1.2 Pin Description

The following table shows MC68HC16Z1 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 the table, MC68HC16Z1 Driver Types, for a description of output drivers. An entry in the discrete I/O column of the MC68HC16Z1 Pin Characteristics table indicates that a pin has an alternate I/O function. The port designation is given when it applies. Refer to the MC68HC16Z1 Block Diagram for information about port organization.

Table 2 MC68HC16Z1 Pin Characteristics

Pin Mnemonic	Output Driver	Input Synchronized	Input Hysteresis	Discrete I/O	Port Designation
ADDR23/$\overline{CS10}$/ECLK	A	Y	N	O	—
ADDR[22:19]/$\overline{CS[9:6]}$	A	Y	N	O	C[6:3]
ADDR[18:0]	A	Y	N	—	—
AN[7:0][1]	—	Y	N	I	ADA[7:0]
$\overline{AS}$	B	Y	N	I/O	E5
$\overline{AVEC}$	B	Y	N	I/O	E2
$\overline{BERR}$	B	Y	N	—	—
$\overline{BG}$/$\overline{CS1}$	B	—	—	—	—
$\overline{BGACK}$/$\overline{CS2}$	B	Y	N	—	—
$\overline{BKPT}$/DSCKL	—	Y	Y	—	—
$\overline{BR}$/$\overline{CS0}$	B	Y	N	O	Separate
CLKOUT	A	—	—	—	—
$\overline{CSBOOT}$	B	—	—	—	—
DATA[15:0][1]	AW	Y	N	—	—
$\overline{DS}$	B	Y	N	I/O	E4
$\overline{DSACK1}$	B	Y	N	I/O	E1
$\overline{DSACK0}$	B	Y	N	I/O	E0
DSI/IPIPE1	A	Y	Y	—	Separate
DSO/IPIPE0	A	—	—	—	Separate
EXTAL[2]	—	—	Special	—	—
FC[2:0]/$\overline{CS[5:3]}$	A	Y	N	O	C[2:0]
FREEZE/QUOT	A	—	—	—	—
$\overline{HALT}$	Bo	Y	N	—	—
IC4/OC5	A	Y	Y	I/O	GP4
IC[3:1]	A	Y	Y	I/O	GP[7:5]
$\overline{IRQ[7:1]}$	B	Y	Y	I/O	F[7:1]
MISO	Bo	Y	Y	I/O	QS0
MODCLK[1]	B	Y	N	I/O	F0
MOSI	Bo	Y	Y	I/O	QS1
OC[4:1]	A	Y	Y	I/O	GP[3:0]
PAI[3]	—	Y	Y	I	Separate
PCLK[3]	—	Y	Y	I	Separate
PCS0/$\overline{SS}$	Bo	Y	Y	I/O	QS3
PCS[3:1]	Bo	Y	Y	I/O	QS[6:4]
PWMA, PWMB[4]	A	—	—	O	Separate

Table 2 MC68HC16Z1 Pin Characteristics (Continued)

Pin Mnemonic	Output Driver	Input Synchronized	Input Hysteresis	Discrete I/O	Port Designation
R/$\overline{W}$	A	Y	N	—	—
$\overline{RESET}$	Bo	Y	Y	—	—
RXD	—	N	N	—	—
SCK	Bo	Y	Y	I/O	QS2
SIZ[1:0]	B	Y	N	I/O	E[7:6]
$\overline{TSTME}$/TSC	—	Y	Y	—	—
TXD	Bo	Y	Y	I/O	QS7
V_{RH}[5]	—	—	—	—	—
V_{RL}[5]	—	—	—	—	—
XFC[2]	—	—	—	Special	—
XTAL[2]	—	—	—	Special	—

NOTES

1. DATA[15:0] are synchronized during reset only. MODCLK, MCCI and ADC pins are synchronized only when used as input port pins.
2. EXTAL, XFC, and XTAL are clock reference connections.
3. PAI and PCLK can be used for discrete input, but are not part of an I/O port.
4. PWMA and PWMB can be used for discrete output, but are not part of an I/O port.
5. V_{RH} and V_{RL} are ADC reference voltage inputs.

Table 3 MC68HC16Z1 Power Connections

V_{STBY}	Standby RAM Power/Clock Synthesizer Power
V_{DDSYN}	Clock Synthesizer Power
V_{DDA}/V_{SSA}	A/D Converter Power
V_{SSE}/V_{DDE}	External Periphery Power (Source and Drain)
V_{SSI}/V_{DDI}	Internal Module Power (Source and Drain)

Table 4 MC68HC16Z1 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

1.3 Signal Description

Use the following tables as a quick reference to MC68HC16Z1 signal type and function.

Table 5 MC68HC16Z1 Signal Characteristics

Signal Name	MCU Module	Signal Type	Active State
ADDR[23:0]	SIM	Bus	—
AN[7:0]	ADC	Input	—
$\overline{AS}$	SIM	Output	0
$\overline{AVEC}$	SIM	Input	0
$\overline{BERR}$	SIM	Input	0
$\overline{BG}$	SIM	Output	0
$\overline{BGACK}$	SIM	Input	0
$\overline{BKPT}$	CPU16	Input	0
$\overline{BR}$	SIM	Input	0
CLKOUT	SIM	Output	—
$\overline{CS[10:0]}$	SIM	Output	0
$\overline{CSBOOT}$	SIM	Output	0
DATA[15:0]	SIM	Bus	—
$\overline{DS}$	SIM	Output	0
$\overline{DSACK[1:0]}$	SIM	Input	0
DSCLK	CPU16	Input	Serial Clock
DSI	CPU16	Input	(Serial Data)
DSO	CPU16	Output	(Serial Data)
EXTAL	SIM	Input	—
FC[2:0]	SIM	Output	—
FREEZE	SIM	Output	1
$\overline{HALT}$	SIM	Input/Output	0
IC[4:1]	GPT	Input	—
IPIPE0	CPU16	Output	—
IPIPE1	CPU16	Output	—
$\overline{IRQ[7:1]}$	SIM	Input	0
MISO	QSM	Input/Output	—
MODCLK	SIM	Input	—
MOSI	QSM	Input/Output	—
OC[5:1]	GPT	Output	—
PADA[7:0]	ADC	Input	(Port)
PAI	GPT	Input	—
PC[6:0]	SIM	Output	(Port)
PE[7:0]	SIM	Input/Output	(Port)
PF[7:0]	SIM	Input/Output	(Port)
PGP[7:0]	GPT	Input/Output	(Port)
PQS[7:0]	QSM	Input/Output	(Port)
PCLK	GPT	Input	—
PCS[3:0]	QSM	Input/Output	—
PWMA, PWMB	GPT	Output	—
QUOT	SIM	Output	—

Table 5 MC68HC16Z1 Signal Characteristics (Continued)

Signal Name	MCU Module	Signal Type	Active State
R/$\overline{W}$	SIM	Output	1/0
$\overline{RESET}$	SIM	Input/Output	0
RXD	QSM	Input	—
SCK	QSM	Input/Output	—
SIZ[1:0]	SIM	Output	—
$\overline{SS}$	QSM	Input	0
TSC	SIM	Input	—
$\overline{TSTME}$	SIM	Input	0
TXD	QSM	Output	—
V_{RH}	ADC	Input	—
V_{RL}	ADC	Input	—
XFC	SIM	Input	—
XTAL	SIM	Output	—

Table 6 MC68HC16Z1 Signal Function

Signal Name	Mnemonic	Function
Address Strobe	$\overline{AS}$	Indicates that a valid address is on the address bus
Autovector	$\overline{AVEC}$	Requests an automatic vector during interrupt acknowledge
Bus Error	$\overline{BERR}$	Indicates that a bus error has occurred
Bus Grant	$\overline{BG}$	Indicates that the MCU has relinquished the bus
Bus Grant Acknowledge	$\overline{BGACK}$	Indicates that an external device has assumed bus mastership
Breakpoint	$\overline{BKPT}$	Signals a hardware breakpoint to the CPU
Bus Request	$\overline{BR}$	Indicates that an external device requires bus mastership
Chip Selects	$\overline{CS[10:0]}$	Select external devices at programmed addresses
Boot Chip Select	$\overline{CSBOOT}$	Chip select for external boot start-up ROM
Address Bus	ADDR[19:0]	20-bit address bus used by CPU16
Address Bus	ADDR[23:20]	4 MSB on IMB, test only, outputs follow ADDR19
ADC Analog Input	AN[7:0]	Inputs to ADC MUX
System Clockout	CLKOUT	System clock output
Data Bus	DATA[15:0]	16-bit data bus
Data Strobe	$\overline{DS}$	During a read cycle, indicates that an external device should place valid data on the data bus. During a write cycle, indicates that valid data is on the data bus
Halt	$\overline{HALT}$	Suspend external bus activity
Interrupt Request Level	$\overline{IRQ[7:1]}$	Provides an interrupt priority level to the CPU
Data and Size Acknowledge	$\overline{DSACK[1:0]}$	Provide asynchronous data transfers and dynamic bus sizing
Peripheral Chip Select	PCS[3:0]	QSPI peripheral chip selects
Reset	$\overline{RESET}$	System reset
Test Mode Enable	$\overline{TSTME}$	Hardware enable for SIM test mode
Development Serial In, Out, Clock	DSI, DSO, DSCLK	Serial I/O and clock for background debug mode
Crystal Oscillator	EXTAL, XTAL	Connections for clock synthesizer circuit reference; a crystal or an external oscillator can be used
Function Codes	FC[2:0]	Identify processor state and current address space

Table 6 MC68HC16Z1 Signal Function (Continued)

Signal Name	Mnemonic	Function
Freeze	FREEZE	Indicates that the CPU has entered background mode
Instruction Pipeline	PIPE[1:0]	Indicate instruction pipeline activity
Master In Slave Out	MISO	Serial input to QSPI in master mode; serial output from QSPI in slave mode
Clock Mode Select	MODCLK	Selects the source and type of system clock
Master Out Slave In	MOSI	Serial output from QSPI in master mode; serial input to QSPI in slave mode
Port ADA	PADA[7:0]	ADC digital input port signals
Port C	PC[6:0]	SIM digital output port signals
Port E	PE[7:0]	SIM digital I/O port signals
Port F	PF[7:0]	SIM digital I/O port signals
Port GP	PGP[7:0]	GPT digital I/O port signals
Port QS	PQS[7:0]	QSM digital I/O port signals
Quotient Out	QUOT	Provides the quotient bit of the polynomial divider
Read/Write	R/$\overline{W}$	Indicates the direction of data transfer on the bus
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	SIZ[1:0]	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	TSC	Places all output drivers in a high-impedance state
SCI Transmit Data	TXD	Serial output from the SCI
ADC Reference Voltage	$V_{rh,Vrl}$	Provide precise reference for A/D conversion
External Filter Capacitor	XFC	Connection for external phase-locked loop filter capacitor
Crystal Oscillator	EXTAL, XTAL	Connections for clock synthesizer circuit reference; a crystal or an external oscillator can be used
Function Codes	FC[2:0]	Identify processor state and current address space
Freeze	FREEZE	Indicates that the CPU has entered background mode
Instruction Pipeline	PIPE[1:0]	Indicate instruction pipeline activity
Master In Slave Out	MISO	Serial input to QSPI in master mode; serial output from QSPI in slave mode
Clock Mode Select	MODCLK	Selects the source and type of system clock
Master Out Slave In	MOSI	Serial output from QSPI in master mode; serial input to QSPI in slave mode
Port ADA	PADA[7:0]	ADC digital input port signals
Port C	PC[6:0]	SIM digital output port signals
Port E	PE[7:0]	SIM digital I/O port signals
Port F	PF[7:0]	SIM digital I/O port signals
Port GP	PGP[7:0]	GPT digital I/O port signals
Port QS	PQS[7:0]	QSM digital I/O port signals
Quotient Out	QUOT	Provides the quotient bit of the polynomial divider
Read/Write	R/$\overline{W}$	Indicates the direction of data transfer on the bus
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	SIZ[1:0]	Indicates the number of bytes to be transferred during a bus cycle

Table 6 MC68HC16Z1 Signal Function (Continued)

Signal Name	Mnemonic	Function
Slave Select	$\overline{SS}$	Causes serial transmission when QSPI is in slave mode; causes mode fault in master mode
Three-State Control	TSC	Places all output drivers in a high-impedance state
SCI Transmit Data	TXD	Serial output from the SCI
ADC Reference Voltage	$V_{rh,Vrl}$	Provide precise reference for A/D conversion
External Filter Capacitor	XFC	Connection for external phase-locked loop filter capacitor

1.4 Internal Register Address Map

In the following figure, IMB ADDR[23:20] are represented by the letter Y. The value represented by Y determines the base address of MCU module control registers. In the MC68HC16Z1, Y is equal to M111, where M is the logic state of the module mapping (MM) bit in the system integration module configuration register (SIMCR). Since the CPU16 uses only ADDR[19:0], and ADDR[23:20] follow the logic state of ADDR19 when CPU driven, the CPU cannot access IMB addresses from $080000 to $F7FFFF. In order for the MCU to function correctly, MM must be set (Y must equal $F). If M is cleared, internal registers are mapped to base address $700000, and are inaccessible until a reset occurs. The SRAM array is positioned by a base address register in the SRAM CTRL block. Unimplemented blocks are mapped externally.

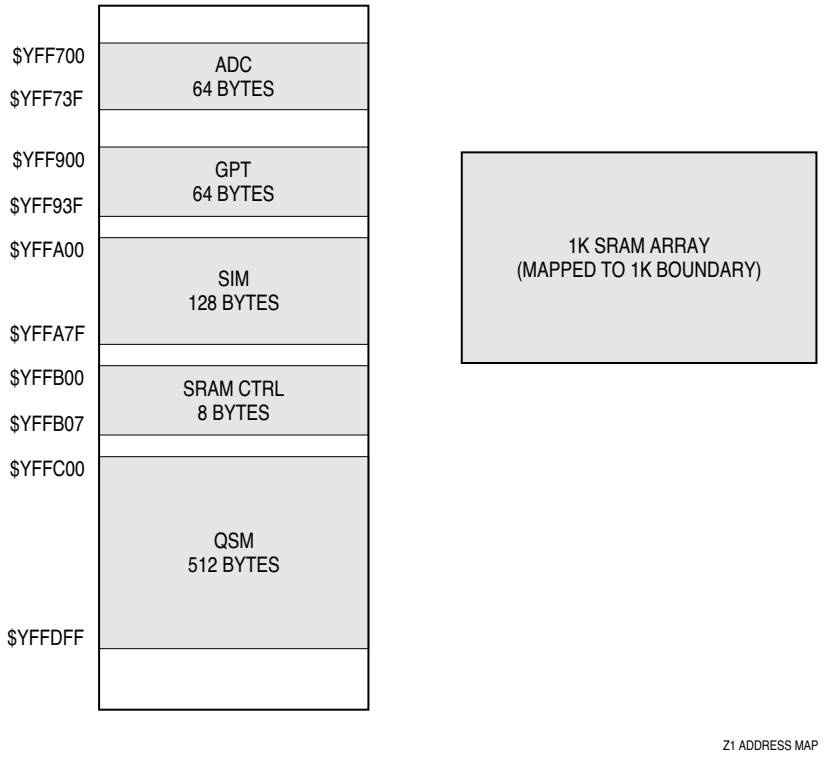


Figure 4 Internal Register Addresses

1.5 Pseudolinear Memory Maps

The following figures both show the complete CPU16 pseudolinear address space. Address space can be split into physically distinct program and data spaces by decoding the MCU function code outputs. The first figure shows the memory map of a system that has combined program and data spaces. The second figure shows the memory map when MCU function code outputs are decoded. Reset and exception vectors are mapped into bank 0 and cannot be relocated. The CPU16 program counter, stack pointer, and Z index register can be initialized to any address in pseudolinear memory, but exception vectors are limited to 16-bit addresses — to access locations outside of bank 0 during exception handler routines (including interrupt exceptions), a jump table must be used.

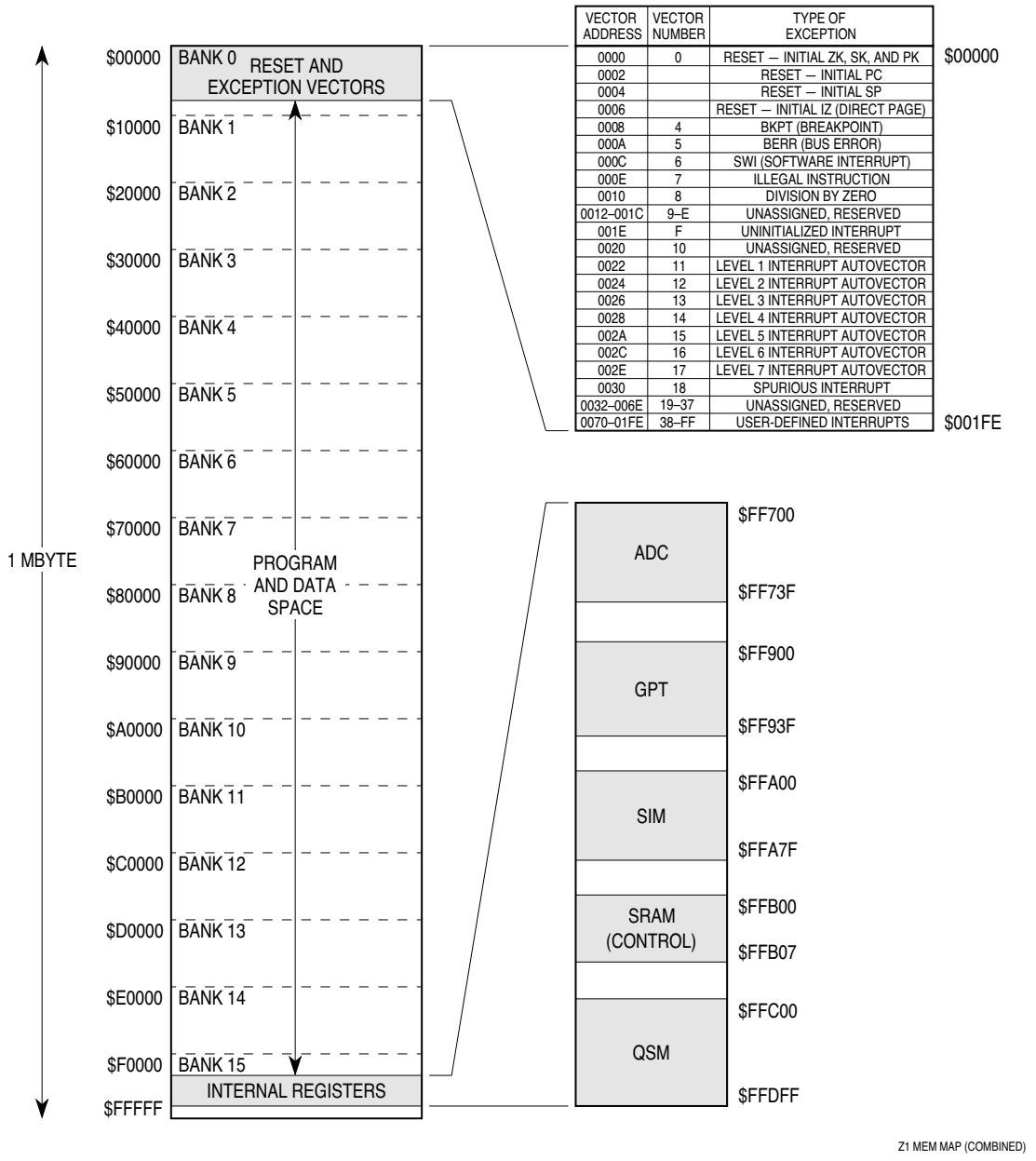


Figure 5 Pseudolinear Addressing With Combined Program and Data Spaces

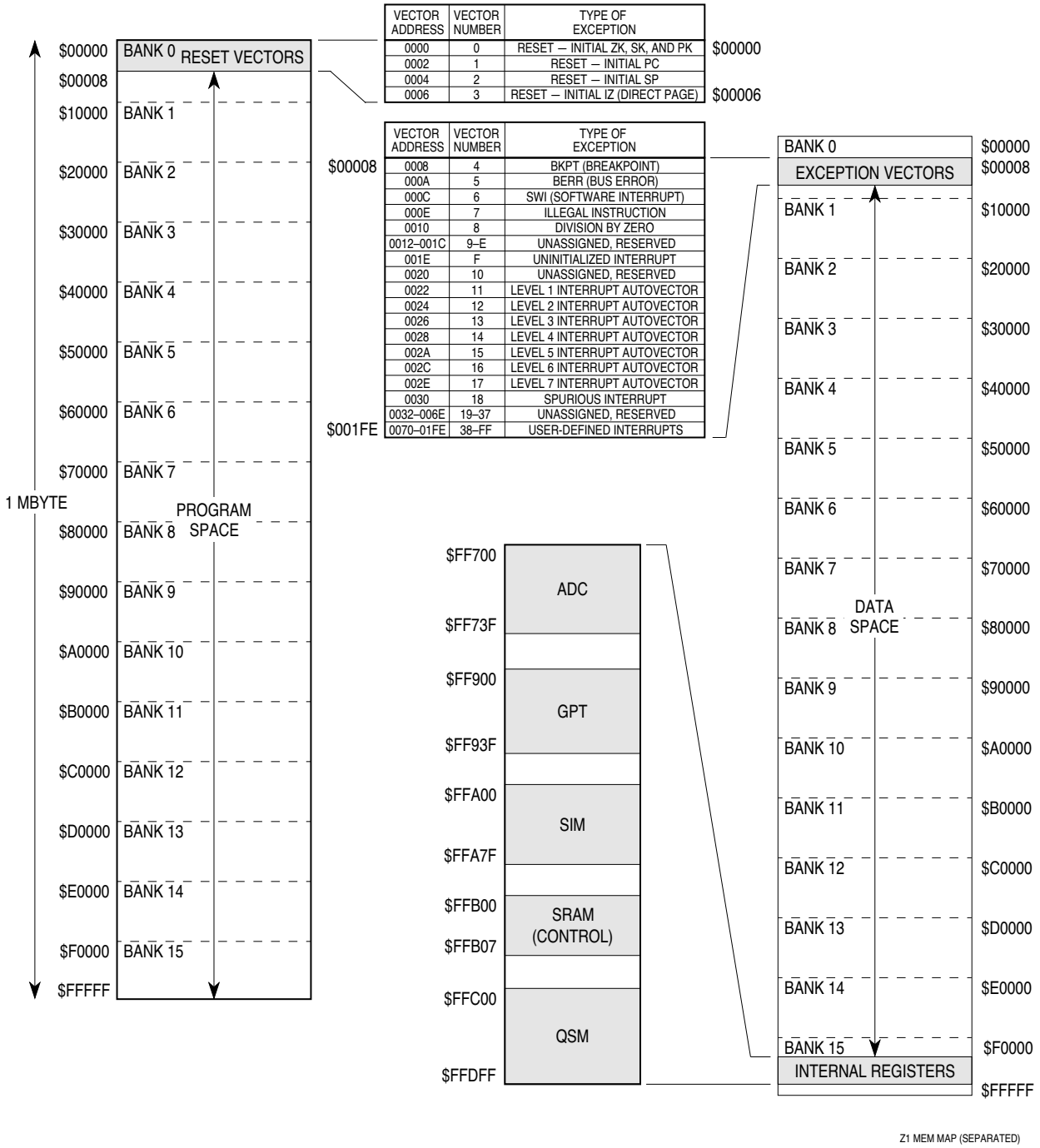


Figure 6 Pseudolinear Addressing With Separated Program and Data Spaces

1.6 Intermodule Bus

The intermodule bus (IMB) is a standardized bus developed to facilitate both design and operation of modular microcontrollers. It contains circuitry to support exception processing, address space partitioning, multiple interrupt levels, and vectored interrupts. The standardized modules in the MC68HC16Z1 communicate with one another and with external components via the IMB. Although the full IMB supports 24 address and 16 data lines, the MC68HC16Z1 uses only 16 data lines and 20 address lines. Because the CPU16 uses only 20 address lines, ADDR[23:20] are tied to ADDR19 when processor driven. ADDR[23:20] are brought out to pins for test purposes.

2 CPU16

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.

2.1 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.

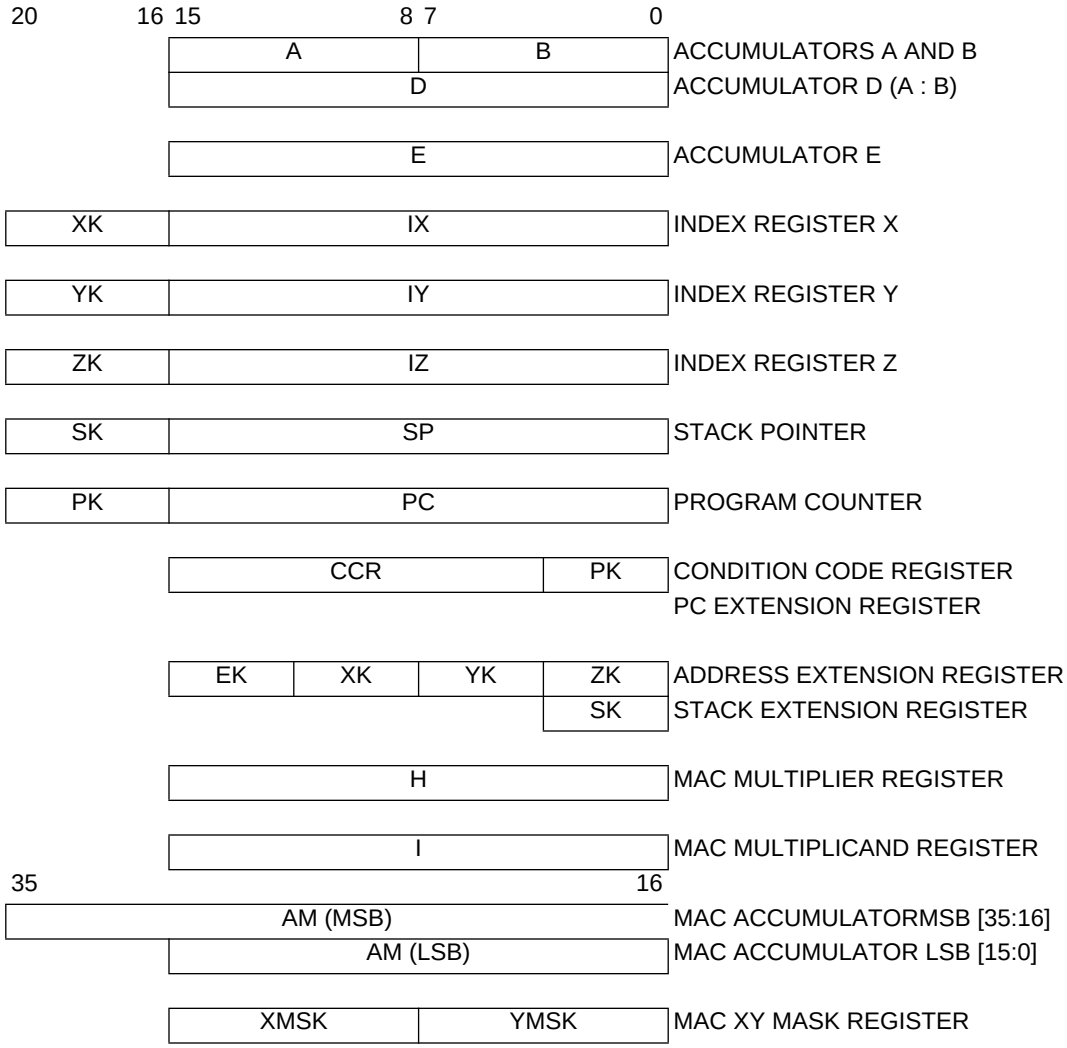
2.2 M68HC11 Compatibility

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.

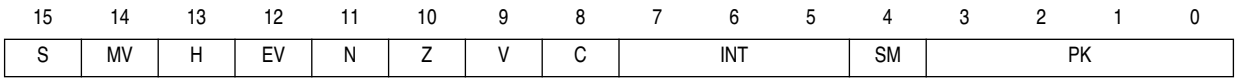
The CPU16 also has several new or enhanced addressing modes. M68HC11 CPU direct mode addressing has been replaced by a special form of indexed addressing that uses the new IZ register and a reset vector to provide greater flexibility.

2.3 Programmer's Model



Accumulator A — 8-bit general-purpose register
Accumulator B — 8-bit general-purpose register
Accumulator D — 16-bit 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

2.3.1 Condition Code Register



The condition code register can be considered as two functional blocks. The MSB, which corresponds to the CCR in the M68HC11, contains the low-power stop control bit and processor status flags. The LSB contains the interrupt priority field, the DSP saturation mode control bit, and the program counter address extension field.

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

MV — Accumulator M overflow flag
Set when overflow into the accumulator M sign bit (AM35) has occurred

H — Half Carry Flag
Set when a carry from bit 3 in accumulators A or B occurs during BCD addition

EV — Extension Bit Overflow Flag
Set when an overflow into bit 31 of accumulator M has occurred

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

Z — Zero Flag
Set when all bits of a result register are zero

V — Overflow Flag
Set when two's complement overflow occurs as the result of an operation

C — Carry Flag
Set when a carry or borrow occurs during arithmetic operation. Also used during shift and rotate operations to facilitate multiple word operations.

INT[2:0] — Interrupt Priority Mask
The value of this field ($0 to $7) specifies the CPU16 interrupt priority level.

SM — Saturate Mode Bit
When SM is set, if either EV or MV is set, data read from accumulator M using TMRT 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 pseudolinear address.

2.4 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.

2.5 Addressing Modes

The CPU16 provides 10 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.

2.6 Instruction Set

The CPU16 has an 8-bit instruction set. It uses a prebyte to support a multipage opcode map. This arrangement makes it possible to fetch an 8-bit operand simultaneously with a Page 0 opcode. If a program makes maximum use of 8-bit offset indexed addressing mode, it will have a significantly smaller instruction space.

The 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.

The following table is a summary of the 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 7 Instruction Set Summary

Mnemonic	Operation	Description	Address Mode	Instruction			Condition Codes							
				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 X	(XK : IX) + (000 : B) ⇒ XK : IX	INH	374F	—	2	—	—	—	—	—	—	—	—
ABY	Add B to Y	(YK : IY) + (000 : B) ⇒ YK : IY	INH	375F	—	2	—	—	—	—	—	—	—	—
ABZ	Add B to Z	(ZK : IZ) + (000 : B) ⇒ ZK : IZ	INH	376F	—	2	—	—	—	—	—	—	—	—
ACE	Add E to AM[31:15]	(AM[31:15]) + (E) ⇒ AM	INH	3722	—	2	—	Δ	—	Δ	—	—	—	—
ACED	Add concatenated E and D to AM	(E : D) + (AM) ⇒ 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								
			E, X	27C3	—	6								
			E, Y	27D3	—	6								
			E, Z	27E3	—	6								
			IND16, X	17C3	gggg	6								
			IND16, Y	17D3	gggg	6								
			IND16, Z	17E3	gggg	6								
			EXT	17F3	hh ll	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								
			E, X	2783	—	6								
			E, Y	2793	—	6								
			E, Z	27A3	—	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								
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								
			E, X	2741	—	6								
			E, Y	2751	—	6								
			E, Z	2761	—	6								
			IND16, X	1741	gggg	6								
			IND16, Y	1751	gggg	6								
			IND16, Z	1761	gggg	6								
			EXT	1771	hh ll	6								

Table 7 Instruction Set Summary (Continued)

Mnemonic	Operation	Description	Address Mode	Instruction			Condition Codes							
				Opcode	Operand	Cycles	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								
			E, X	27C1	—	6								
			E, Y	27D1	—	6								
			E, Z	27E1	—	6								
			IND16, X	17C1	gggg	6								
			IND16, Y	17D1	gggg	6								
			IND16, Z	17E1	gggg	6								
			EXT	17F1	hh ll	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								
			E, X	2781	—	6								
			E, Y	2791	—	6								
			E, Z	27A1	—	6								
			IMM16	37B1	jjkk	4								
			IND16, X	37C1	gggg	6								
			IND16, Y	37D1	gggg	6								
			IND16, Z	37E1	gggg	6								
			EXT	37F1	hh ll	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 X	(XK : IX) + («D) ⇒ XK : IX	INH	37CD	—	2	—	—	—	—	—	—	—	—
ADY	Add D to Y	(YK : IY) + («D) ⇒ YK : IY	INH	37DD	—	2	—	—	—	—	—	—	—	—
ADZ	Add D to Z	(ZK : IZ) + («D) ⇒ ZK : IZ	INH	37ED	—	2	—	—	—	—	—	—	—	—
AEX	Add E to X	(XK : IX) + («E) ⇒ XK : IX	INH	374D	—	2	—	—	—	—	—	—	—	—
AEY	Add E to Y	(YK : IY) + («E) ⇒ YK : IY	INH	375D	—	2	—	—	—	—	—	—	—	—
AEZ	Add E to Z	(ZK : IZ) + («E) ⇒ ZK : IZ	INH	376D	—	2	—	—	—	—	—	—	—	—
AIS	Add Immediate Data to SP	SK : SP + «IMM ⇒ SK : SP	IMM8	3F	ii	2	—	—	—	—	—	—	—	—
			IMM16	373F	jj kk	4								
AIX	Add Immediate Value to X	XK : IX + «IMM ⇒ XK : IX	IMM8	3C	ii	2	—	—	—	—	—	Δ	—	—
			IMM16	373C	jj kk	4								
AIY	Add Immediate Value to Y	YK : IY + «IMM ⇒ YK : IY	IMM8	3D	ii	2	—	—	—	—	—	Δ	—	—
			IMM16	373D	jj kk	4								
AIZ	Add Immediate Value to Z	ZK : IZ + «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								
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								

Table 7 Instruction Set Summary (Continued)

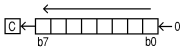
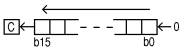
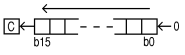
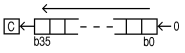
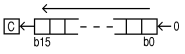
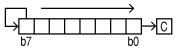
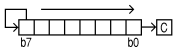
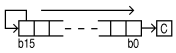
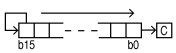
Mnemonic	Operation	Description	Address Mode	Instruction			Condition Codes							
				Opcode	Operand	Cycles	S	MV	H	EV	N	Z	V	C
ANDD	AND D	(D) • (M : M + 1) ⇒ D	IND8, X	86	ff	6	—	—	—	—	Δ	Δ	0	—
			IND8, Y	96	ff	6								
			IND8, Z	A6	ff	6								
			E, X	2786	—	6								
			E, Y	2796	—	6								
			E, Z	27A6	—	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								
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
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

Table 7 Instruction Set Summary (Continued)

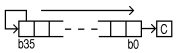
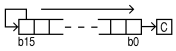
Mnemonic	Operation	Description	Address Mode	Instruction			Condition Codes							
				Opcode	Operand	Cycles	S	MV	H	EV	N	Z	V	C
ASRM	Arithmetic Shift Right AM		INH	27BA	—	4	—	—	—	Δ	Δ	—	—	Δ
ASRW	Arithmetic Shift Right Word		IND16, X IND16, Y IND16, Z EXT	270D 271D 272D 273D	gggg gggg gggg hh ll	8 8 8 8	—	—	—	—	Δ	Δ	Δ	Δ
BCC[4]	Branch if Carry Clear	If C = 0, branch	REL8	B4	rr	6, 2	—	—	—	—	—	—	—	—
BCLR	Clear Bit(s)	(M) • ($\overline{\text{Mask}}$) ⇒ M	IND16, X IND16, Y IND16, Z EXT IND8, X IND8, Y IND8, Z	08 18 28 38 1708 1718 1728	mm gggg mm gggg mm gggg mm hh ll mm ff mm ff mm ff	8 8 8 8 8 8 8	—	—	—	—	Δ	Δ	0	—
BCLRW	Clear Bit(s) Word	(M : M + 1) • ($\overline{\text{Mask}}$) ⇒ 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[4]	Branch if Carry Set	If C = 1, branch	REL8	B5	rr	6, 2	—	—	—	—	—	—	—	—
BEQ[4]	Branch if Equal	If Z = 1, branch	REL8	B7	rr	6, 2	—	—	—	—	—	—	—	—
BGE[4]	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 enter BDM; else, illegal instruction	INH	37A6	—	—	—	—	—	—	—	—	—	—
BGT [4]	Branch if Greater Than Zero	If Z + (N ⊕ V) = 0, branch	REL8	BE	rr	6, 2	—	—	—	—	—	—	—	—
BHI [4]	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	—
BLE [4]	Branch if Less Than or Equal to Zero	If Z + (N ⊕ V) = 1, branch	REL8	BF	rr	6, 2	—	—	—	—	—	—	—	—
BLS[4]	Branch if Lower or Same	If C + Z = 1, branch	REL8	B3	rr	6, 2	—	—	—	—	—	—	—	—
BLT[4]	Branch if Less Than Zero	If N ⊕ V = 1, branch	REL8	BD	rr	6, 2	—	—	—	—	—	—	—	—
BMI [4]	Branch if Minus	If N = 1, branch	REL8	BB	rr	6, 2	—	—	—	—	—	—	—	—
BNE [4]	Branch if Not Equal	If Z = 0, branch	REL8	B6	rr	6, 2	—	—	—	—	—	—	—	—

Table 7 Instruction Set Summary (Continued)

Mnemonic	Operation	Description	Address Mode	Instruction			Condition Codes							
				Opcode	Operand	Cycles	S	MV	H	EV	N	Z	V	C
BPL[4]	Branch if Plus	If N = 0, branch	REL8	BA	rr	6, 2	—	—	—	—	—	—	—	—
BRA	Branch Always	If 1 = 1, branch	REL8	B0	rr	6	—	—	—	—	—	—	—	—
BRCLR[4]	Branch if Bit(s) Clear	If (M) • (Mask) = 0, branch	IND8, X	CB	mm ff rr	10, 12	—	—	—	—	—	—	—	—
			IND8, Y	DB	mm ff rr	10, 12								
			IND8, Z	EB	mm ff rr	10, 12								
			IND16, X	0A	mm gggg rrrr	10, 14								
			IND16, Y	1A	mm gggg rrrr	10, 14								
			IND16, Z	2A	mm gggg rrrr	10, 14								
			EXT	3A	mm hh ll rrrr	10, 14								
BRN	Branch Never	If 1 = 0, branch	REL8	B1	rr	2	—	—	—	—	—	—	—	—
BRSET[4]	Branch if Bit(s) Set	If (M) • (Mask) = 0, branch	IND8, X	8B	mm ff rr	10, 12	—	—	—	—	—	—	—	—
			IND8, Y	9B	mm ff rr	10, 12								
			IND8, Z	AB	mm ff rr	10, 12								
			IND16, X	0B	mm gggg rrrr	10, 14								
			IND16, Y	1B	mm gggg rrrr	10, 14								
			IND16, Z	2B	mm gggg rrrr	10, 14								
			EXT	3B	mm hh ll rrrr	10, 14								
BSET	Set Bit(s)	(M) • (Mask) ⇒ M	IND16, X	09	mm gggg	8	—	—	—	—	Δ	Δ	0	—
			IND16, Y	19	mm gggg	8								
			IND16, Z	29	mm gggg	8								
			EXT	39	mm hh ll	8								
			IND8, X	1709	mm ff	8								
			IND8, Y	1719	mm ff	8								
			IND8, Z	1729	mm ff	8								
BSETW	Set Bit(s) in Word	(M : M + 1) • (Mask) ⇒ M : M + 1	IND16, X	2709	gggg mmmm	10	—	—	—	—	Δ	Δ	0	—
			IND16, Y	2719	gggg mmmm	10								
			IND16, Z	2729	gggg mmmm	10								
			EXT	2739	hh ll mmmm	10								
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[4]	Branch if Overflow Clear	If V = 0, branch	REL8	B8	rr	6, 2	—	—	—	—	—	—	—	—
BVS[4]	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 Memory	$00 ⇒ M	IND8, X	05	ff	4	—	—	—	—	0	1	0	0
			IND8, Y	15	ff	4								
			IND8, Z	25	ff	4								
			IND16, X	1705	gggg	6								
			IND16, Y	1715	gggg	6								
			IND16, Z	1725	gggg	6								
			EXT	1735	hh ll	6								
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[32:0]	INH	27B7	—	2	—	0	—	0	—	—	—	—

Table 7 Instruction Set Summary (Continued)

Mnemonic	Operation	Description	Address Mode	Instruction			Condition Codes							
				Opcode	Operand	Cycles	S	MV	H	EV	N	Z	V	C
CLRW	Clear Memory Word	$0000 ⇒ M : M + 1	IND16, X	2705	gggg	6	—	—	—	—	0	1	0	0
			IND16, Y	2715	gggg	6								
			IND16, Z	2725	gggg	6								
			EXT	2735	hh ll	6								
CMPA	Compare A to Memory	(A) – (M)	IND8, X	48	ff	6	—	—	—	—	Δ	Δ	Δ	Δ
			IND8, Y	58	ff	6								
			IND8, Z	68	ff	6								
			IMM8	78	ii	2								
			IND16, X	1748	gggg	6								
			IND16, Y	1758	gggg	6								
			IND16, Z	1768	gggg	6								
			EXT	1778	hh ll	6								
			E, X	2748	—	6								
			E, Y	2758	—	6								
			E, Z	2768	—	6								
CMPB	Compare B to Memory	(B) – (M)	IND8, X	C8	ff	6	—	—	—	—	Δ	Δ	Δ	Δ
			IND8, Y	D8	ff	6								
			IND8, Z	E8	ff	6								
			IMM8	F8	ii	2								
			IND16, X	17C8	gggg	6								
			IND16, Y	17D8	gggg	6								
			IND16, Z	17E8	gggg	6								
			EXT	17F8	hh ll	6								
			E, X	27C8	—	6								
			E, Y	27D8	—	6								
			E, Z	27E8	—	6								
COM	One's Complement	$FF – (M) ⇒ M	IND8, X	00	ff	8	—	—	—	—	Δ	Δ	0	1
			IND8, Y	10	ff	8								
			IND8, Z	20	ff	8								
			IND16, X	1700	gggg	8								
			IND16, Y	1710	gggg	8								
			IND16, Z	1720	gggg	8								
			EXT	1730	hh ll	8								
COMA	One's Complement A	$FF – (A) ⇒ A	INH	3700	—	2	—	—	—	—	Δ	Δ	0	1
COMB	One's Complement B	$FF – (B) ⇒ B	INH	3710	—	2	—	—	—	—	Δ	Δ	0	1
COMD	One's Complement D	$FFFF – (D) ⇒ D	INH	27F0	—	2	—	—	—	—	Δ	Δ	0	1
COME	One's Complement E	$FFFF – (E) ⇒ E	INH	2770	—	2	—	—	—	—	Δ	Δ	0	1
COMW	One's Complement Word	$FFFF – M : M + 1 ⇒ M : M + 1	IND16, X	2700	gggg	8	—	—	—	—	Δ	Δ	0	1
			IND16, Y	2710	gggg	8								
			IND16, Z	2720	gggg	8								
			EXT	2730	hh ll	8								
CPD	Compare D to Memory	(D) – (M : M + 1)	IND8, X	88	ff	6	—	—	—	—	Δ	Δ	Δ	Δ
			IND8, Y	98	ff	6								
			IND8, Z	A8	ff	6								
			E, X	2788	—	6								
			E, Y	2798	—	6								
			E, Z	27A8	—	6								
			IMM16	37B8	jj kk	4								
			IND16, X	37C8	gggg	6								
			IND16, Y	37D8	gggg	6								
			IND16, Z	37E8	gggg	6								
			EXT	37F8	hh ll	6								
CPE	Compare E to Memory	(E) – (M : M + 1)	IMM16	3738	jjkk	4	—	—	—	—	Δ	Δ	Δ	Δ
			IND16, X	3748	gggg	6								
			IND16, Y	3758	gggg	6								
			IND16, Z	3768	gggg	6								
			EXT	3778	hhll	6								
CPS	Compare SP 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								
			IND16, X	174F	gggg	6								
			IND16, Y	175F	gggg	6								
			IND16, Z	176F	gggg	6								
			EXT	177F	hh ll	6								
			IMM16	377F	jj kk	4								

Table 7 Instruction Set Summary (Continued)

Mnemonic	Operation	Description	Address Mode	Instruction			Condition Codes							
				Opcode	Operand	Cycles	S	MV	H	EV	N	Z	V	C
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								
			IND16, X	174C	gggg	6								
			IND16, Y	175C	gggg	6								
			IND16, Z	176C	gggg	6								
			EXT	177C	hh ll	6								
			IMM16	377C	jj kk	4								
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								
			IND16, X	174D	gggg	6								
			IND16, Y	175D	gggg	6								
			IND16, Z	176D	gggg	6								
			EXT	177D	hh ll	6								
			IMM16	377D	jj kk	4								
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								
			IND16, X	174E	gggg	6								
			IND16, Y	175E	gggg	6								
			IND16, Z	176E	gggg	6								
			EXT	177E	hh ll	6								
			IMM16	377E	jj kk	4								
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 Divide	(E : D) / (IX) Quotient ⇒ IX Remainder ⇒ D	INH	3728	—	24	—	—	—	—	Δ	Δ	Δ	Δ
EDIVS	Extended Signed Divide	(E : D) / (IX) Quotient ⇒ IX Remainder ⇒ ACCD	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 7 Instruction Set Summary (Continued)

Mnemonic	Operation	Description	Address Mode	Instruction			Condition Codes							
				Opcode	Operand	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								
			E, X	2784	—	6								
			E, Y	2794	—	6								
			E, Z	27A4	—	6								
			IMM16	37B4	jjkk	4								
			IND16, X	37C4	gggg	6								
			IND16, Y	37D4	gggg	6								
			IND16, Z	37E4	gggg	6								
			EXT	37F4	hhll	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	IND20, X	4B	zg gggg	8	—	—	—	—	—	—	—	—
			IND20, Y	5B	zg gggg	8								
			IND20, Z	6B	zg gggg	8								
			EXT20	7A	zb hh ll	6								
JSR	Jump to Subroutine	Push (PC) (SK : SP) – 2 ⇒ SK : SP Push (CCR) (SK : SP) – 2 ⇒ SK : SP ⟨ea⟩ ⇒ PK : PC	IND20, X	89	zg gggg	12	—	—	—	—	—	—	—	—
			IND20, Y	99	zg gggg	12								
			IND20, Z	A9	zg gggg	12								
			EXT20	FA	zb hh ll	10								
LBCC[4]	Long Branch if Carry Clear	If C = 0, branch	REL16	3784	rrrr	6, 4	—	—	—	—	—	—	—	—
LBCS[4]	Long Branch if Carry Set	If C = 1, branch	REL16	3785	rrrr	6, 4	—	—	—	—	—	—	—	—
LBEQ[4]	Long Branch if Equal	If Z = 1, branch	REL16	3787	rrrr	6, 4	—	—	—	—	—	—	—	—
LBEV[4]	Long Branch if EV Set	If EV = 1, branch	REL16	3791	rrrr	6, 4	—	—	—	—	—	—	—	—
LBGE[4]	Long Branch if Greater Than or Equal to Zero	If N ⊕ V = 0, branch	REL16	378C	rrrr	6, 4	—	—	—	—	—	—	—	—
LBGT[4]	Long Branch if Greater Than Zero	If Z ✣ (N ⊕ V) = 0, branch	REL16	378E	rrrr	6, 4	—	—	—	—	—	—	—	—

Table 7 Instruction Set Summary (Continued)

Mnemonic	Operation	Description	Address Mode	Instruction			Condition Codes							
				Opcode	Operand	Cycles	S	MV	H	EV	N	Z	V	C
LBHI [4]	Long Branch if Higher	If C ✣ Z = 0, branch	REL16	3782	rrrr	6, 4	—	—	—	—	—	—	—	—
LBLE [4]	Long Branch if Less Than or Equal to Zero	If Z ✣ (N ⊕ V) = 1, branch	REL16	378F	rrrr	6, 4	—	—	—	—	—	—	—	—
LBLS [4]	Long Branch if Lower or Same	If C ✣ Z = 1, branch	REL16	3783	rrrr	6, 4	—	—	—	—	—	—	—	—
LBLT [4]	Long Branch if Less Than Zero	If N ⊕ V = 1, branch	REL16	378D	rrrr	6, 4	—	—	—	—	—	—	—	—
LBMI [4]	Long Branch if Minus	If N = 1, branch	REL16	378B	rrrr	6, 4	—	—	—	—	—	—	—	—
LBMV [4]	Long Branch if MV Set	If MV = 1, branch	REL16	3790	rrrr	6, 4	—	—	—	—	—	—	—	—
LBNE [4]	Long Branch if Not Equal	If Z = 0, branch	REL16	3786	rrrr	6, 4	—	—	—	—	—	—	—	—
LBPL [4]	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 [4]	Long Branch if Overflow Clear	If V = 0, branch	REL16	3788	rrrr	6, 4	—	—	—	—	—	—	—	—
LBVS [4]	Long Branch if Overflow Set	If V = 1, branch	REL16	3789	rrrr	6, 4	—	—	—	—	—	—	—	—
LDAA	Load A	(M) ⇒ A	IND8, X	45	ff	6	—	—	—	—	Δ	Δ	0	—
			IND8, Y	55	ff	6								
			IND8, Z	65	ff	6								
			IMM8	75	ii	2								
			IND16, X	1745	gggg	6								
			IND16, Y	1755	gggg	6								
			IND16, Z	1765	gggg	6								
			EXT	1775	hh ll	6								
			E, X	2745	—	6								
			E, Y	2755	—	6								
			E, Z	2765	—	6								
LDAB	Load B	(M) ⇒ B	IND8, X	C5	ff	6	—	—	—	—	Δ	Δ	0	—
			IND8, Y	D5	ff	6								
			IND8, Z	E5	ff	6								
			IMM8	F5	ii	2								
			IND16, X	17C5	gggg	6								
			IND16, Y	17D5	gggg	6								
			IND16, Z	17E5	gggg	6								
			EXT	17F5	hh ll	6								
			E, X	27C5	—	6								
			E, Y	27D5	—	6								
			E, Z	27E5	—	6								
LDD	Load D	(M : M + 1) ⇒ D	IND8, X	85	ff	6	—	—	—	—	Δ	Δ	0	—
			IND8, Y	95	ff	6								
			IND8, Z	A5	ff	6								
			E, X	2785	—	6								
			E, Y	2795	—	6								
			E, Z	27A5	—	6								
			IMM16	37B5	jj kk	4								
			IND16, X	37C5	gggg	6								
			IND16, Y	37D5	gggg	6								
			IND16, Z	37E5	gggg	6								
			EXT	37F5	hh ll	6								
LDE	Load E	(M : M + 1) ⇒ E	IMM16	3735	jj kk	4	—	—	—	—	Δ	Δ	0	—
			IND16, X	3745	gggg	6								
			IND16, Y	3755	gggg	6								
			IND16, Z	3765	gggg	6								
			EXT	3775	hh ll	6								
LDED	Load Concatenated E and D	(M : M + 1) ⇒ E (M + 2 : M + 3) ⇒ D	EXT	2771	hh ll	8	—	—	—	—	—	—	—	—

Table 7 Instruction Set Summary (Continued)

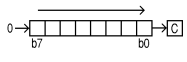
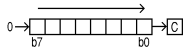
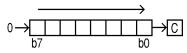
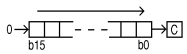
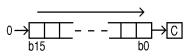
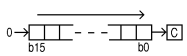
Mnemonic	Operation	Description	Address Mode	Instruction			Condition Codes							
				Opcode	Operand	Cycles	S	MV	H	EV	N	Z	V	C
LDHI	Initialize H and I	$(M : M + 1)_X \Rightarrow H\ R$ $(M : M + 1)_Y \Rightarrow I\ R$	EXT	27B0	—	8	—	—	—	—	—	—	—	—
LDS	Load SP	$(M : M + 1) \Rightarrow SP$	IND8, X IND8, Y 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 IND16, X IND16, Y IND16, Z EXT IMM16	CC DC EC 17CC 17DC 17EC 17FC 37BC	ff ff ff gggg gggg gggg hh ll jj kk	6 6 6 6 6 6 6 4	—	—	—	—	Δ	Δ	0	—
LDY	Load IY	$(M : M + 1) \Rightarrow IY$	IND8, X IND8, Y IND8, Z IND16, X IND16, Y IND16, Z EXT IMM16	CD DD ED 17CD 17DD 17ED 17FD 37BD	ff ff ff gggg gggg gggg hh ll jj kk	6 6 6 6 6 6 6 4	—	—	—	—	Δ	Δ	0	—
LDZ	Load IZ	$(M : M + 1) \Rightarrow IZ$	IND8, X IND8, Y IND8, Z IND16, X IND16, Y IND16, Z EXT IMM16	CE DE EE 17CE 17DE 17EE 17FE 37BE	ff ff ff gggg gggg gggg hh ll jj kk	6 6 6 6 6 6 6 4	—	—	—	—	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 7 Instruction Set Summary (Continued)

Mnemonic	Operation	Description	Address Mode	Instruction			Condition Codes							
				Opcode	Operand	Cycles	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	30	ff hh ll	8	—	—	—	—	Δ	Δ	0	—
			EXT to IXP	32	ff hh ll	8								
			EXT to EXT	37FE	hh ll hh ll	10								
MOVW	Move Word	$(M : M + 1_1) \Rightarrow M : M + 1_2$	IXP to EXT	31	ff hh ll	8	—	—	—	—	Δ	Δ	0	—
			EXT to IXP	33	ff hh ll	8								
			EXT to EXT	37FF	hh ll hh ll	10								
MUL	Multiply	(A) ∗ (B) ⇒ D	INH	3724	—	10	—	—	—	—	—	—	—	Δ
NEG	Negate Memory	$00 – (M) ⇒ M	IND8, X	02	ff	8	—	—	—	—	Δ	Δ	Δ	Δ
			IND8, Y	12	ff	8								
			IND8, Z	22	ff	8								
			IND16, X	1702	gggg	8								
			IND16, Y	1712	gggg	8								
			IND16, Z	1722	gggg	8								
			EXT	1732	hh ll	8								
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	2702	gggg	8	—	—	—	—	Δ	Δ	Δ	Δ
			IND16, Y	2712	gggg	8								
			IND16, Z	2722	gggg	8								
			EXT	2732	hh ll	8								
NOP	Null Operation	—	INH	274C	—	2	—	—	—	—	—	—	—	—
ORAA	OR A	(A) ✣ (M) ⇒ A	IND8, X	47	ff	6	—	—	—	—	Δ	Δ	0	—
			IND8, Y	57	ff	6								
			IND8, Z	67	ff	6								
			IMM8	77	ii	2								
			IND16, X	1747	gggg	6								
			IND16, Y	1757	gggg	6								
			IND16, Z	1767	gggg	6								
			EXT	1777	hh ll	6								
			E, X	2747	—	6								
			E, Y	2757	—	6								
			E, Z	2767	—	6								
ORAB	OR B	(B) ✣ (M) ⇒ B	IND8, X	C7	ff	6	—	—	—	—	Δ	Δ	0	—
			IND8, Y	D7	ff	6								
			IND8, Z	E7	ff	6								
			IMM8	F7	ii	2								
			IND16, X	17C7	gggg	6								
			IND16, Y	17D7	gggg	6								
			IND16, Z	17E7	gggg	6								
			EXT	17F7	hh ll	6								
			E, X	27C7	—	6								
			E, Y	27D7	—	6								
			E, Z	27E7	—	6								
ORD	OR D	(D) ✣ (M : M + 1) ⇒ D	IND8, X	87	ff	6	—	—	—	—	Δ	Δ	0	—
			IND8, Y	97	ff	6								
			IND8, Z	A7	ff	6								
			E, X	2787	—	6								
			E, Y	2797	—	6								
			E, Z	27A7	—	6								
			IMM16	37B7	jj kk	4								
			IND16, X	37C7	gggg	6								
			IND16, Y	37D7	gggg	6								
			IND16, Z	37E7	gggg	6								
			EXT	37F7	hh ll	6								

Table 7 Instruction Set Summary (Continued)

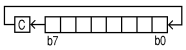
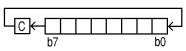
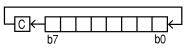
Mnemonic	Operation	Description	Address Mode	Instruction			Condition Codes							
				Opcode	Operand	Cycles	S	MV	H	EV	N	Z	V	C
ORE	OR E	(E) ✣ (M : M + 1) ⇒ E	IMM16 IND16, X IND16, Y IND16, Z EXT	3737 3747 3757 3767 3777	jj kk gggg gggg gggg hh ll	4 6 6 6 6	—	—	—	—	Δ	Δ	0	—
ORP [1]	OR Condition Code Register	(CCR) ✣ IMM16 ⇒ CCR	IMM16	373B	jj kk	4	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ
PSHA	Push A	(SK : SP) + 1 ⇒ SK : SP Push (A) (SK : SP) – 2 ⇒ SK : SP	INH	3708	—	4	—	—	—	—	—	—	—	—
PSHB	Push B	(SK : SP) + 1 ⇒ SK : SP Push (B) (SK : SP) – 2 ⇒ 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 State	MAC Registers ⇒ Stack	INH	27B8	—	14	—	—	—	—	—	—	—	—
PULA	Pull A	(SK : SP) + 2 ⇒ SK : SP Pull (A) (SK : SP) – 1 ⇒ SK : SP	INH	3709	—	6	—	—	—	—	—	—	—	—
PULB	Pull B	(SK : SP) + 2 ⇒ SK : SP Pull (B) (SK : SP) – 1 ⇒ 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; $(M : M + 1)_X \Rightarrow H$; $(M : M + 1)_Y \Rightarrow I$ (E) – 1 ⇒ E	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 7 Instruction Set Summary (Continued)

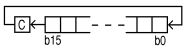
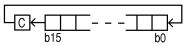
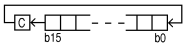
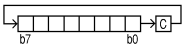
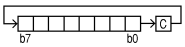
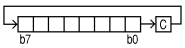
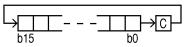
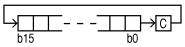
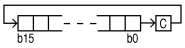
Mnemonic	Operation	Description	Address Mode	Instruction			Condition Codes							
				Opcode	Operand	Cycles	S	MV	H	EV	N	Z	V	C
ROLD	Rotate Left D		INH	27FC	—	2	—	—	—	—	Δ	Δ	Δ	Δ
ROLE	Rotate Left E		INH	277C	—	2	—	—	—	—	Δ	Δ	Δ	Δ
ROLW	Rotate Left Word		IND16, X IND16, Y IND16, Z EXT	270C 271C 272C 273C	gggg gggg gggg hh ll	8 8 8 8	—	—	—	—	Δ	Δ	Δ	Δ
ROR	Rotate Right		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[2]	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[3]	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 7 Instruction Set Summary (Continued)

Mnemonic	Operation	Description	Address Mode	Instruction			Condition Codes							
				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								
			E, X	2782	—	6								
			E, Y	2792	—	6								
			E, Z	27A2	—	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								
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	6	—	—	—	—	Δ	Δ	0	—
			IND8, Y	9A	ff	6								
			IND8, Z	AA	ff	6								
			E, X	278A	—	6								
			E, Y	279A	—	6								
			E, Z	27AA	—	6								
			IND16, X	37CA	gggg	4								
			IND16, Y	37DA	gggg	4								
			IND16, Z	37EA	gggg	4								
			EXT	37FA	hh ll	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 7 Instruction Set Summary (Continued)

Mnemonic	Operation	Description	Address Mode	Instruction			Condition Codes							
				Opcode	Operand	Cycles	S	MV	H	EV	N	Z	V	C
STS	Store SP	(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								
			E, X	2780	—	6								
			E, Y	2790	—	6								
			E, Z	27A0	—	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								
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 7 Instruction Set Summary (Continued)

Mnemonic	Operation	Description	Address Mode	Instruction			Condition Codes							
				Opcode	Operand	Cycles	S	MV	H	EV	N	Z	V	C
SWI	Software Interrupt	(PK : PC) + 2 ⇒ PK : PC Push (PC) (SK : SP) – 2 ⇒ SK : SP Push (CCR) (SK : SP) – 2 ⇒ SK : SP $0 ⇒ PK SWI Vector ⇒ PC	INH	3720	—	16	—	—	—	—	—	—	—	—
SXT	Sign Extend B into A	If B7 = 1 then A = $FF else A = $00	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) ⇒ EK	INH	27FA	—	2	—	—	—	—	—	—	—	—
TBSK	Transfer B to SK	(B) ⇒ SK	INH	379F	—	2	—	—	—	—	—	—	—	—
TBXK	Transfer B to XK	(B) ⇒ XK	INH	379C	—	2	—	—	—	—	—	—	—	—
TBYK	Transfer B to YK	(B) ⇒ YK	INH	379D	—	2	—	—	—	—	—	—	—	—
TBZK	Transfer B to ZK	(B) ⇒ 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	(D) ⇒ AM[15:0] (E) ⇒ AM[31:16] AM[35:32] = AM31	INH	27B1	—	4	—	0	—	0	—	—	—	—
TEKB	Transfer EK to B	$0 ⇒ B[7:4] (EK) ⇒ B[3:0]	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 AM to E Rounded	Rounded (AM) ⇒ Temp If (SM • (EV ✣ MV)) then Saturation ⇒ E else Temp[31:16] ⇒ E	INH	27B4	—	6	—	Δ	—	Δ	Δ	Δ	—	—
TMET	Transfer AM to E Truncated	If (SM • (EV ✣ MV)) then Saturation ⇒ 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 MSB 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 for 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 7 Instruction Set Summary (Continued)

Mnemonic	Operation	Description	Address Mode	Instruction			Condition Codes							
				Opcode	Operand	Cycles	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 X	(SK : SP) + 2 ⇒ XK : IX	INH	274F	—	2	—	—	—	—	—	—	—	—
TSY	Transfer SP to Y	(SK : SP) + 2 ⇒ YK : IY	INH	275F	—	2	—	—	—	—	—	—	—	—
TSZ	Transfer SP to Z	(SK : SP) + 2 ⇒ ZK : IZ	INH	276F	—	2	—	—	—	—	—	—	—	—
TXKB	Transfer XK to B	$0 ⇒ B[7:4] (XK) ⇒ B[3:0]	INH	37AC	—	2	—	—	—	—	—	—	—	—
TXS	Transfer X to SP	(XK : IX) – 2 ⇒ SK : SP	INH	374E	—	2	—	—	—	—	—	—	—	—
TXY	Transfer X to Y	(XK : IX) ⇒ YK : IY	INH	275C	—	2	—	—	—	—	—	—	—	—
TXZ	Transfer X to Z	(XK : IX) ⇒ ZK : IZ	INH	276C	—	2	—	—	—	—	—	—	—	—
TYKB	Transfer YK to B	$0 ⇒ B[7:4] (YK) ⇒ B[3:0]	INH	37AD	—	2	—	—	—	—	—	—	—	—
TYS	Transfer Y to SP	(YK : IY) – 2 ⇒ SK : SP	INH	375E	—	2	—	—	—	—	—	—	—	—
TYX	Transfer Y to X	(YK : IY) ⇒ XK : IX	INH	274D	—	2	—	—	—	—	—	—	—	—
TYZ	Transfer Y to Z	(YK : IY) ⇒ ZK : IZ	INH	276D	—	2	—	—	—	—	—	—	—	—
TZKB	Transfer ZK to B	$0 ⇒ B[7:4] (ZK) ⇒ B[3:0]	INH	37AE	—	2	—	—	—	—	—	—	—	—
TZS	Transfer Z to SP	(ZK : IZ) – 2 ⇒ SK : SP	INH	376E	—	2	—	—	—	—	—	—	—	—
TZX	Transfer Z to X	(ZK : IZ) ⇒ XK : IX	INH	274E	—	2	—	—	—	—	—	—	—	—
TZY	Transfer Z to Y	(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 X	(D) ⇔ (IX)	INH	37CC	—	2	—	—	—	—	—	—	—	—
XGDY	Exchange D with Y	(D) ⇔ (IY)	INH	37DC	—	2	—	—	—	—	—	—	—	—
XGDZ	Exchange D with Z	(D) ⇔ (IZ)	INH	37EC	—	2	—	—	—	—	—	—	—	—
XGEX	Exchange E with X	(E) ⇔ (IX)	INH	374C	—	2	—	—	—	—	—	—	—	—
XGEY	Exchange E with Y	(E) ⇔ (IY)	INH	375C	—	2	—	—	—	—	—	—	—	—
XGEZ	Exchange E with Z	(E) ⇔ (IZ)	INH	376C	—	2	—	—	—	—	—	—	—	—

NOTES:

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

Table 8 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 (2'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

2.7 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.

2.7.1 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 instruction vector table, which is located in the first 512 bytes of bank 0.

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 CPU16 left shifts the vector number one place (multiplies by two) to convert it to an address.

Table 9 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	BKPT (Breakpoint)
5	000A	D	BERR (Bus Error)
6	000C	D	SWI (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

2.7.2 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. The following figure shows the exception stack frame.

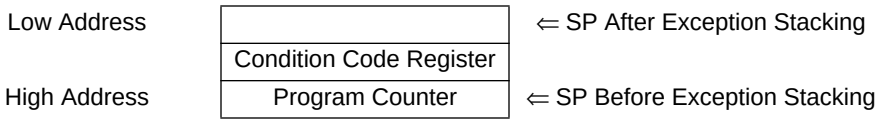


Figure 7 Exception Stack Frame

2.7.3 Exception Processing Sequence

Exception processing is performed in four distinct phases.

- Priority of all pending exceptions is evaluated, and the highest priority exception is processed first.
- Processor state is stacked, then the CCR PK extension field is cleared.
- An exception vector number is acquired and converted to a vector address.
- 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 reset 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.

2.7.4 Types of Exceptions

Exceptions can be generated either internally or externally. External exceptions which are defined as asynchronous, include interrupts, bus errors (BERR), breakpoints (BKPT), and resets (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. Refer to **3 System Integration Module** for more information about resets and interrupts.

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

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

Because of pipelining, the stacked return PK : PC value for 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 in order to resume execution of the interrupted instruction stream. 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. Since RTI always subtracts $0006 upon return, the stacked PK : PC must be adjusted by the instruction that caused the exception so that execution will resume with the following instruction. $0002 is added to the PK : PC value before it is stacked.

2.7.5 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 done by priority, from lowest to highest. Note that priority governs the order in which exception processing occurs, not the order in which exception handlers are executed.

Unless bus error, breakpoint, or reset occur 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 BERR exception processing, for example, the first instruction of the BERR handler is executed before interrupts are sensed. This permits the exception handler to mask interrupts during execution.

2.7.6 RTI Instruction

The return-from-interrupt instruction (RTI) must be the last instruction in all exception handlers except for 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 a reset initializes the stack pointer and does not create a stack frame.

3 System Integration Module

The MC68HC16Z1 system integration module (SIM) consists of five functional blocks that control system start-up, initialization, configuration, and the external bus.

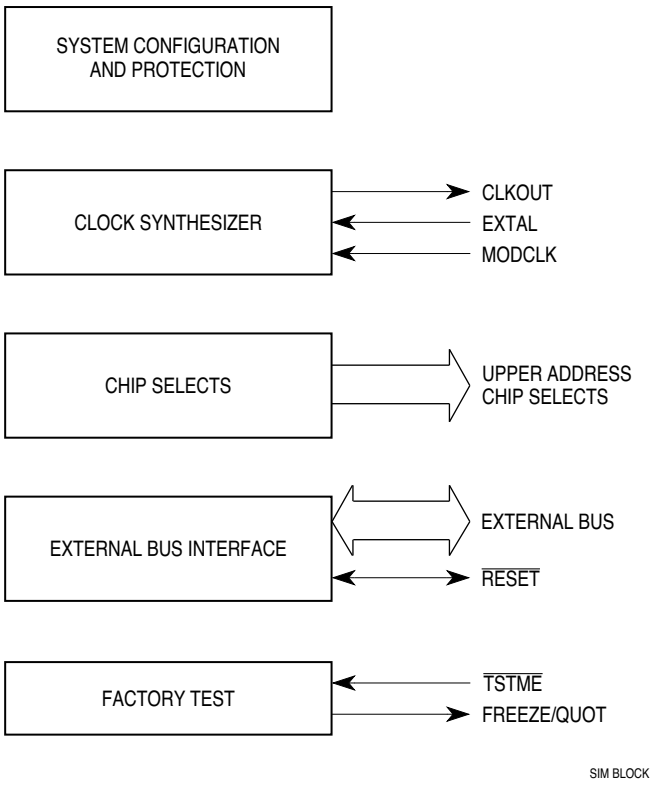


Figure 8 SIM Block Diagram

Table 10 SIM Address Map

Address	15 8	7 0
YFFA00	MODULE CONFIGURATION (SIMCR)	
YFFA02	FACTORY TEST (SIMTR)	
YFFA04	CLOCK SYNTHESIZER CONTROL (SYNCR)	
YFFA06	UNUSED	RESET STATUS REGISTER (RSR)
YFFA08	MODULE TEST E (SIMTRE)	
YFFA0A	UNUSED	UNUSED
YFFA0C	UNUSED	UNUSED
YFFA0E	UNUSED	UNUSED
YFFA10	UNUSED	PORTE DATA (PORTE0)
YFFA12	UNUSED	PORTE DATA (PORTE1)
YFFA14	UNUSED	PORTE DATA DIRECTION (DDRE)
YFFA16	UNUSED	PORTE PIN ASSIGNMENT (PEPAR)
YFFA18	UNUSED	PORTF DATA (PORTF0)
YFFA1A	UNUSED	PORTF DATA (PORTF1)
YFFA1C	UNUSED	PORTF DATA DIRECTION (DDRF)
YFFA1E	UNUSED	PORTF PIN ASSIGNMENT (PFPAR)
YFFA20	UNUSED	SYSTEM PROTECTION CONTROL (SYPCR)
YFFA22	PERIODIC INTERRUPT CONTROL (PICR)	
YFFA24	PERIODIC INTERRUPT TIMING (PITR)	
YFFA26	UNUSED	SOFTWARE SERVICE (SWSR)
YFFA28	UNUSED	UNUSED
YFFA2A	UNUSED	UNUSED
YFFA2C	UNUSED	UNUSED
YFFA2E	UNUSED	UNUSED
YFFA30	TEST MODULE MASTER SHIFT A (TSTMSRA)	
YFFA32	TEST MODULE MASTER SHIFT B (TSTMSRB)	
YFFA34	TEST MODULE SHIFT COUNT (TSTSC)	
YFFA36	TEST MODULE REPETITION COUNTER (TSTRC)	
YFFA38	TEST MODULE CONTROL (CREG)	
YFFA3A	TEST MODULE DISTRIBUTED REGISTER (DREG)	
YFFA3C	UNUSED	UNUSED
YFFA3E	UNUSED	UNUSED
YFFA40	UNUSED	PORT C DATA (PORTC)
YFFA42	UNUSED	UNUSED
YFFA44	CHIP-SELECT PIN ASSIGNMENT (CSPAR0)	
YFFA46	CHIP-SELECT PIN ASSIGNMENT (CSPAR1)	
YFFA48	CHIP-SELECT BASE BOOT (CSBARBT)	
YFFA4A	CHIP-SELECT OPTION BOOT (CSORBT)	
YFFA4C	CHIP-SELECT BASE 0 (CSBAR0)	
YFFA4E	CHIP-SELECT OPTION 0 (CSOR0)	
YFFA50	CHIP-SELECT BASE 1 (CSBAR1)	
YFFA52	CHIP-SELECT OPTION 1 (CSOR1)	
YFFA54	CHIP-SELECT BASE 2 (CSBAR2)	
YFFA56	CHIP-SELECT OPTION 2 (CSOR2)	
YFFA58	CHIP-SELECT BASE 3 (CSBAR3)	
YFFA5A	CHIP-SELECT OPTION 3 (CSOR3)	
YFFA5C	CHIP-SELECT BASE 4 (CSBAR4)	

Table 10 SIM Address Map

Address	15 8	7 0
YFFA5E	CHIP-SELECT OPTION 4 (CSOR4)	
YFFA60	CHIP-SELECT BASE 5 (CSBAR5)	
YFFA62	CHIP-SELECT OPTION 5 (CSOR5)	
YFFA64	CHIP-SELECT BASE 6 (CSBAR6)	
YFFA66	CHIP-SELECT OPTION 6 (CSOR6)	
YFFA68	CHIP-SELECT BASE 7 (CSBAR7)	
YFFA6A	CHIP-SELECT OPTION 7 (CSOR7)	
YFFA6C	CHIP-SELECT BASE 8 (CSBAR8)	
YFFA6E	CHIP-SELECT OPTION 8 (CSOR8)	
YFFA70	CHIP-SELECT BASE 9 (CSBAR9)	
YFFA72	CHIP-SELECT OPTION 9 (CSOR9)	
YFFA74	CHIP-SELECT BASE 10 (CSBAR10)	
YFFA76	CHIP-SELECT OPTION 10 (CSOR10)	
YFFA78	UNUSED	UNUSED
YFFA7A	UNUSED	UNUSED
YFFA7C	UNUSED	UNUSED
YFFA7E	UNUSED	UNUSED

Y = M111 where M is the logic state of the modmap (MM) bit in the SIMCR

3.1 System Configuration and Protection

This functional block provides configuration control for the entire MC68HC16Z1. It also performs interrupt arbitration, bus monitoring, and system test functions.

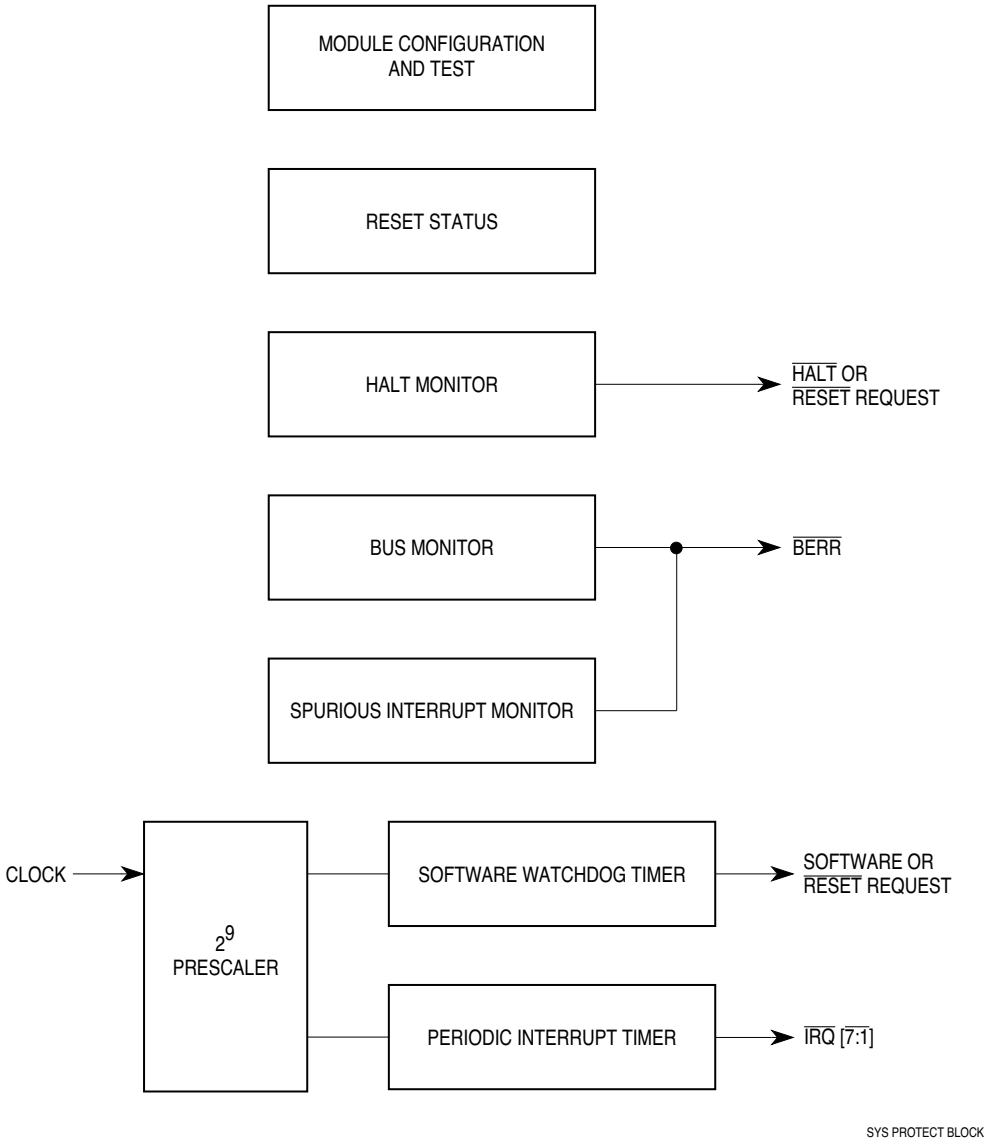


Figure 9 System Configuration and Protection Block Diagram

3.2 System Configuration

The SIM controls M68HC16 configuration during normal operation and during internal testing.

MCR — Module Configuration Register **$YFFA00**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXOFF	FRZSW	FRZBM	0	SLVEN	0	SHEN		SUPV	MM	0	0	IARB			
RESET:															
0	0	0	0	DB11	0	0	0	1	1	0	0	1	1	1	1

The module configuration register controls system configuration. It can be read or written at any time, except for the module mapping (MM) bit, which can be written once and must remain set.

EXOFF — External Clock Off

0 = The CLKOUT pin is driven from an internal clock source.

1 = The CLKOUT pin is placed in a high-impedance state.

FRZSW — Freeze Software Enable

0 = When FREEZE is asserted, the software watchdog and periodic interrupt timer counters continue to run.

1 = When FREEZE is asserted, the software watchdog and periodic interrupt timer counters are disabled, preventing interrupts during software debug.

FRZBM — Freeze Bus Monitor Enable

0 = When FREEZE is asserted, the bus monitor continues to operate.

1 = When FREEZE is asserted, the bus monitor is disabled.

SLVEN — Factory Test Mode Enabled

This bit is a read-only status bit that reflects the state of DB11 during reset.

0 = IMB is not available to an external master.

1 = An external bus master has direct access to the IMB.

SHEN[1:0] — Show Cycle Enable

This field determines what the EBI does with the external bus during internal transfer operations. A show cycle allows internal transfers to be externally monitored. The table below shows whether show cycle data is driven externally, and whether external bus arbitration can occur. To prevent bus conflict, external peripherals must not be enabled during show cycles.

SHEN	Action
00	Show cycles disabled, external arbitration enabled
01	Show cycles enabled, external arbitration disabled
10	Show cycles enabled, external arbitration enabled
11	Show cycles enabled, external arbitration enabled, internal activity is halted by a bus grant

SUPV — Supervisor/Unrestricted Data Space

The SUPV bit places the SIM global registers in either supervisor data space or user data space. The CPU16 in the MC68HC16Z1 operates only in supervisory 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 unless externally driven. MM corresponds to IMB ADDR23. If it is cleared, the SIM maps IMB modules into address space $7FF000 – $7FFFFF, which is inaccessible to the CPU. Modules remain inaccessible until reset occurs. MM can be written once. Initialization software should set MM to logic level 1.

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 SIM routes external interrupt requests to the CPU16, the SIM IARB field value is used for arbitration between internal and external interrupts of the same priority. The reset value of IARB for the SIM is %1111 (highest priority), and the reset IARB value for all other modules is %0000, which prevents SIM interrupts from being discarded during initialization.

3.3 System Protection

MC68HC16Z1 system protection includes a bus monitor, a HALT monitor, a spurious interrupt monitor, and a software watchdog timer. These functions have been made integral to the microcontroller to reduce the number of external components in a complete control system.

The system protection control register controls system monitor functions, software watchdog clock prescaling, and bus monitor timing. In operating mode, this register can be written only once following power-on or reset, but can be read at any time. In test mode, it can be written at any time.

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

7	6	5	4	3	2	1	0
SWE	SWP	SWT		HME	BME	BMT	
RESET:							
1	$\overline{\text{MODCLK}}$	0	0	0	0	0	0

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

SWP — Software Watchdog Prescale

This bit controls the value of the software watchdog prescaler.
- 0 = Software watchdog clock not prescaled
- 1 = Software watchdog clock prescaled by 512

SWT[1:0] — Software Watchdog Timing

This field selects the divide ratio used to establish software watchdog time-out period. The following table gives the ratio for each combination of SWP and SWT bits.

SWP	SWT	Ratio
0	00	2^9
0	01	2^{11}
0	10	2^{13}
0	11	2^{15}
1	00	2^{18}
1	01	2^{20}
1	10	2^{22}
1	11	2^{24}

HME — Halt Monitor Enable
- 0 = Disable halt monitor function
- 1 = Enable halt monitor function

BME — Bus Monitor External Enable
- 0 = Disable bus monitor function for an internal to external bus cycle.
- 1 = Enable bus monitor function for an internal to external bus cycle.

BMT[1:0] — Bus Monitor Timing

This field selects a bus monitor time-out period as shown in the following table.

BMT	Bus Monitor Time-out Period
00	64 System Clocks (CLK)
01	32 System Clocks (CLK)
10	16 System Clocks (CLK)
11	8 System Clocks (CLK)

3.3.1 Bus Monitor

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

$\overline{\text{DSACKx}}$ and $\overline{\text{AVEC}}$ response times are measured in clock cycles. The maximum allowable response time can be selected by setting the BMT field.

The monitor does not check $\overline{\text{DSACKx}}$ response on the external bus unless the CPU16 initiates the bus cycle. The BME bit in the SYPCR enables the internal bus monitor for internal to external bus cycles. 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.

3.3.2 Halt Monitor

The halt monitor responds to an assertion of $\overline{\text{HALT}}$ on the internal bus. 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 the HME bit in the SYPCR.

3.3.3 Spurious Interrupt Monitor

The spurious interrupt monitor issues $\overline{\text{BERR}}$ if no interrupt arbitration occurs during IACK cycle.

3.3.4 Software Watchdog

SWSR — Software Service Register **$YFFA27**

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0
RESET:							
0	0	0	0	0	0	0	0

Register shown with read value

The software watchdog 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. This register can be written at any time, but returns zeros when read.

Perform a software watchdog service sequence as follows:

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

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

Watchdog clock rate is affected by SWP and SWT in SYPCR. When SWT[1:0] are modified, a watchdog service sequence must be performed before the new time-out period takes effect.

The reset value of SWP is affected by the state of the MODCLK pin on the rising edge of reset, as shown in the following table.

MODCLK	SWP
0	1
1	0

Software watchdog time-out period is given in the following equation:

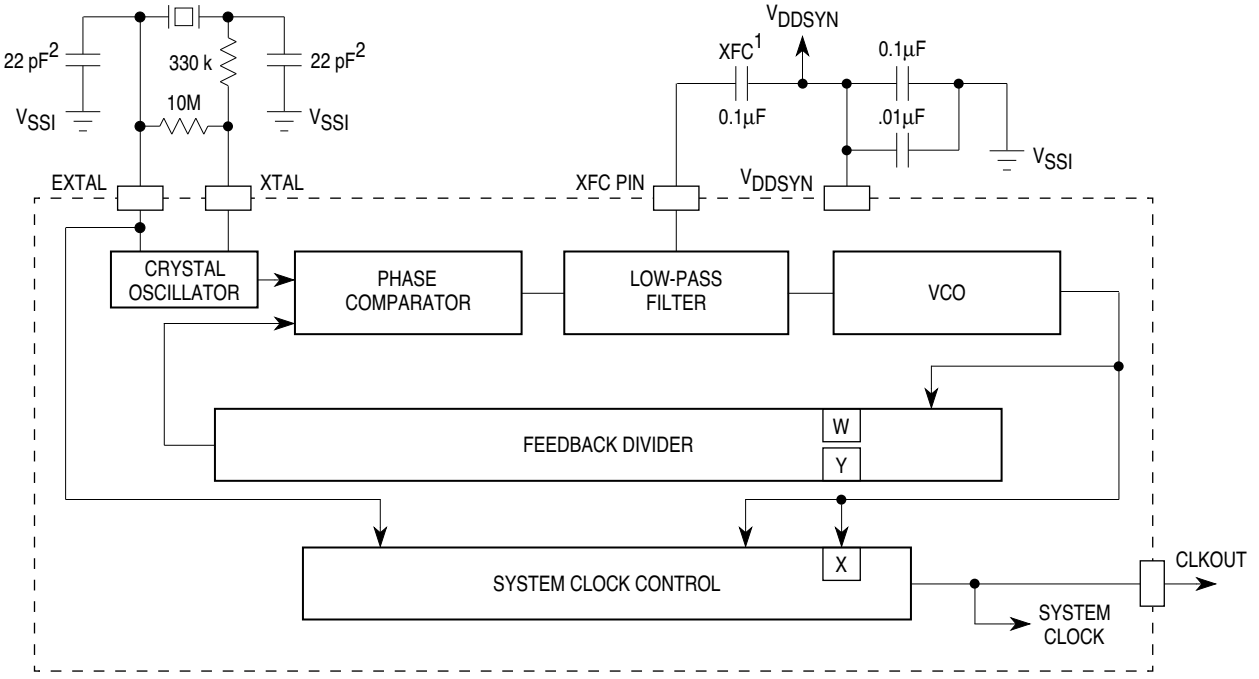
Time-out Period = Divide Count/EXTAL Frequency

3.4 System Clock

The system clock in the SIM provides timing signals for the IMB modules and for an external peripheral bus. Because the MC68HC16Z1 is a fully static design, register and memory contents are not affected when clock rate changes. System hardware and software support changes in clock rate during operation.

The system clock signal can be generated in three ways. An internal phase-locked loop can synthesize the clock from an internal frequency source, an external frequency source, or the clock signal can be input from an external source.

Following is a block diagram of the clock submodule.



1. Must be low-leakage capacitor (insulation resistance 30,000 MΩ or greater).
2. Capacitance based on a test circuit constructed with a DAISHINKU DMX-38 32.768-kHz crystal.

Figure 10 System Clock Block Diagram

3.4.1 Clock Sources

The state of the clock mode (MODCLK) pin during reset determines clock source. When MODCLK is held high during reset, the clock synthesizer generates a clock signal from either a crystal oscillator or an external reference input. Clock synthesizer control register SYNCR determines operating frequency and various modes of operation. When MODCLK is held low during reset, the clock synthesizer is disabled, and an external system clock signal must be applied. When the synthesizer is disabled, SYNCR control bits have no effect.

A reference crystal must be connected between the EXTAL and XTAL pins in order to use the internal oscillator. A 32.768-kHz watch crystal is recommended — these crystals are readily available and inexpensive. MC68HC16Z1 clock synthesizer specifications are based upon a typical 32.768-kHz crystal.

If an external reference signal or an external system clock signal is applied through the EXTAL pin, the XTAL pin must be left floating. External reference signal frequency must be less than or equal to maximum specified reference frequency. External system clock signal frequency must be less than or equal to maximum specified system clock frequency.

When an external system clock signal is applied (PLL not used), 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:

$$\text{Minimum external clock period} = \frac{\text{minimum external clock high/low time}}{\text{50\% – percentage variation of external clock input duty cycle}}$$

3.4.2 Clock Synthesizer Operation

A voltage controlled oscillator (VCO) generates the system clock signal. A portion of 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 internal oscillator or from an external source. The comparator generates a control signal proportional to the difference in phase between its two inputs. The signal is low-pass filtered and used to correct VCO output frequency.

The synthesizer locks when VCO frequency is identical to reference frequency. Lock time is affected by the filter time constant and by the amount of difference between the two comparator inputs. Whenever comparator input changes, the synthesizer must re-lock. Lock status is shown by the SLOCK bit in SYNCR.

The MC68HC16Z1 does not come out of reset state until the synthesizer locks. Crystal type, characteristic frequency, and layout of external oscillator circuitry affect lock time.

The low-pass filter requires an external low-leakage capacitor, typically 0.1 μF, connected between the XFC and V_{DDSYN} pins.

V_{DDSYN} is used to power the clock circuits. A separate power source increases MCU noise immunity and can be used to run the clock when the MCU is powered down. Use a quiet power supply as the V_{DDSYN} source, since PLL stability depends on the VCO, which uses this supply. Place adequate external bypass capacitors as close as possible to the V_{DDSYN} pin to ensure stable operating frequency.

When the clock synthesizer is used, control register SYNCR determines operating frequency and various modes of operation. Because the CPU16 in the MC68HC16Z1 operates only in supervisor mode, SYNCR can be read or written at any time.

The SYNCR X bit controls a divide by two prescaler that is not in the synthesizer feedback loop. Setting X doubles clock speed without changing VCO speed. There is no VCO relock delay. The SYNCR W bit controls a 3-bit prescaler in the feedback divider. Setting W increases VCO speed by a factor of four. The SYNCR Y field determines the count modulus for a modulo 64 down counter, causing it to divide by a value of Y + 1. When either W or Y value changes, there is a VCO relock delay.

Clock frequency is determined by SYNCR bit settings as follows:

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

In order 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.

VCO frequency is determined by:

$$F_{VCO} = F_{SYSTEM}\,(2 - X)$$

The reset state of SYNCR ($3F00) produces a modulus-64 count — system frequency is 256 times reference frequency.

3.4.3 Clock Control

The clock control circuits determine system clock frequency and clock operation under special circumstances, such as loss of synthesizer reference or low-power mode. Clock source is determined by the logic state of the MODCLK pin during reset.

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
W	X	Y						EDIV	0	0	SLIMP	SLOCK	RSTEN	STSIM	STEXT
RESET:															
0	0	1	1	1	1	1	1	0	0	0	U	U	0	0	0

When the on-chip clock synthesizer is used, system clock frequency is controlled by the bits in the upper byte of SYNCR. Bits in the lower byte show status of or control operation of internal and external clocks. Because the CPU16 always operates in supervisor mode, SYNCR can be read or written at any time.

W — Frequency Control (VCO)

This bit controls a prescaler tap in the synthesizer feedback loop. Setting the bit increases the VCO speed by a factor of four. VCO relock delay is required.

X — Frequency Control Bit (Prescale)

This bit controls a divide by two prescaler that is not in the synthesizer feedback loop. Setting the bit doubles clock speed without changing the VCO speed. There is no VCO relock delay.

Y[5:0] — Frequency Control (Counter)

The Y field controls the modulus down counter in the synthesizer feedback loop, causing it to divide by a value of Y + 1. Values range from 0 to 63. VCO relock delay is required.

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. Refer to **3.5.13 Chip Selects** for more information.

SLIMP — Limp Mode Flag

0 = External crystal is VCO reference.
1 = Loss of crystal reference.

When the on-chip synthesizer is used, loss of reference frequency causes SLIMP to be set. The VCO continues to run using the base control voltage. Maximum limp frequency is maximum specified system clock frequency. X-bit state affects limp frequency.

SLOCK — Synthesizer Lock Flag

0 = VCO is enabled, but has not locked.
1 = VCO has locked on the desired frequency (or system clock is external).

The MCU maintains reset state until the synthesizer locks, but SLOCK does not indicate synthesizer lock status until after the user writes to SYNCR.

RSTEN — Reset Enable

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

STSIM — Stop Mode System Integration Clock

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

STEXT — Stop Mode External Clock

0 = When LPSTOP is executed, the CLKOUT signal is held negated to conserve power.
1 = When LPSTOP is executed, the CLKOUT signal is driven from the SIM clock, as determined by the state of the STSIM bit.

3.4.4 Periodic Interrupt Timer

The periodic interrupt timer (PIT) generates interrupts of specified priorities at specified intervals. Timing for the PIT is provided by a programmable prescaler driven by the system clock.

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

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

This register contains information concerning periodic interrupt priority and vectoring. Bits [10:0] can be read or written at any time. Bits [15:11] are unimplemented and always return zero.

PIRQL[2:0] — Periodic Interrupt Request Level

The following table shows what interrupt request level is asserted when a periodic interrupt is generated. If a PIT interrupt and an external IRQ of the same priority occur simultaneously, the PIT interrupt is serviced first. The periodic timer continues to run when the interrupt is disabled.

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

PIV[7:0] — Periodic Interrupt Vector

The bits of this field contain the vector generated in response to an interrupt from the periodic timer. When the SIM responds, the periodic interrupt vector is placed on the bus.

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	PTP	PITM							
RESET:															
0	0	0	0	0	0	0	$\overline{\text{MODCLK}}$	0	0	0	0	0	0	0	0

PITR contains the count value for the periodic timer. A zero value turns off the periodic timer. This register can be read or written at any time.

PTP — Periodic Timer Prescaler Control

1 = Periodic timer clock prescaled by a value of 512

0 = Periodic timer clock not prescaled

The reset state of PTP is the complement of the state of the MODCLK signal during reset.

PITM[7:0] — Periodic Interrupt Timing Modulus Field

This is an 8-bit timing modulus. The period of the timer can be calculated as follows:

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

where

PIT Period = Periodic interrupt timer period

PITM = Periodic interrupt timer register modulus (PITR[7:0])

EXTAL = Crystal frequency

Prescaler = 512 or 1 depending on the state of the PTP bit in the PITR

3.5 External Bus Interface

The external bus interface (EBI) transfers information between the internal MCU bus and external devices when the MC68HC16Z1 is operating in expanded modes. In fully expanded mode, the external bus has 24 address lines and 16 data lines. In partially expanded mode, the external bus has 24 address lines and 8 data lines. Because the CPU16 in the MC68HC16Z1 drives only 20 of the 24 IMB address lines, ADDR[23:20] follow the output state of ADDR19.

The EBI provides dynamic sizing between 8-bit and 16-bit data accesses. It supports byte, word, and long-word transfers. Ports are accessed through the use of asynchronous cycles controlled by the data transfer (SIZ1 and SIZ0) and data size acknowledge pins ($\overline{\text{DSACK1}}$ and $\overline{\text{DSACK0}}$). In fully expanded mode, both 8-bit and 16-bit data ports can be accessed; in partially expanded mode, only 8-bit ports can be accessed. Multiple bus cycles may be required for a transfer to an 8-bit port.

Port width is the maximum number of bits accepted or provided during a bus transfer. External devices must follow the handshake protocol described below. Control signals indicate the beginning of the cycle, the address space, the size of the transfer, and the type of cycle. The selected device controls the length of the cycle. Strobe signals, one for the address bus and another for the data bus, indicate the validity of an address and provide timing information for data. The EBI operates in an asynchronous mode for any port width.

To add flexibility and minimize the necessity for external logic, MCU chip select logic can be synchronized with EBI transfers. Chip select logic can also provide internally-generated bus control signals for these accesses. Refer to **3.5.13 Chip Selects** for more information.

3.5.1 Bus Control Signals

The CPU initiates a bus cycle by driving the address, size, function code, and read/write outputs. At the beginning of the cycle, size signals SIZ0 and SIZ1 are driven along with the function code signals. The size signals indicate the number of bytes remaining to be transferred during an operand cycle. They are valid while the address strobe ($\overline{\text{AS}}$) is asserted. The following table shows SIZ0 and SIZ1 encoding. The read/write (R/$\overline{\text{W}}$) 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 $\overline{\text{AS}}$ is asserted. R/$\overline{\text{W}}$ 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.

Table 11 Size Signal Encoding

SIZ1	SIZ0	Transfer Size
0	1	Byte
1	0	Word
1	1	3 Byte
0	0	Long 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 to 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 $\overline{\text{AS}}$ is asserted.

Table 12 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[19:0] 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. Because the CPU16 in the MC68HC16Z1 does not drive ADDR[23:20], these lines follow the logic state of ADDR19.

3.5.4 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.

3.5.5 Data Bus

Data bus signals DATA[15:0] comprise a bidirectional, non-multiplexed 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.

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 Bus Cycle Termination Signals

During bus cycles, external devices assert the data transfer and size acknowledge signals ($\overline{DSACK1}$ and $\overline{DSACK0}$). During a read cycle, the signals tell the MCU to terminate the bus cycle and to latch data. During a write cycle, the signals indicate that an external device has successfully stored data and that the cycle can end. These signals also indicate to the MCU the size of the port for the bus cycle just completed. (Refer to the discussion of dynamic bus sizing.)

The bus error ($\overline{BERR}$) signal is also a bus cycle termination indicator and can be used in the absence of $\overline{DSACK1}$ and $\overline{DSACK0}$ to indicate a bus error condition. It can also be asserted in conjunction with these signals, provided it meets the appropriate timing requirements. The internal bus monitor can be used to generate the $\overline{BERR}$ signal for internal and internal-to-external transfers. When $\overline{BERR}$ and $\overline{HALT}$ are asserted simultaneously, the CPU16 takes a bus error exception.

Autovector signal ($\overline{AVEC}$) can terminate external IRQ pin interrupt acknowledge cycles. $\overline{AVEC}$ indicates that the MCU will internally generate a vector number to locate an interrupt handler routine. If it is continuously asserted, autovectors will be generated for all external interrupt requests. $\overline{AVEC}$ is ignored during all other bus cycles.

3.5.8 Data Transfer Mechanism

The MCU architecture supports byte, word, and long-word operands, allowing access to 8- and 16-bit data ports through the use of asynchronous cycles controlled by the data transfer and size acknowledge inputs (DSACK1and $\overline{DSACK0}$).

3.5.9 Dynamic Bus Sizing

The MCU dynamically interprets the port size of the addressed device during each bus cycle, allowing operand transfers to or from 8- and 16-bit ports. During an operand transfer cycle, the slave device signals its port size and indicates completion of the bus cycle to the MCU through the use of the $\overline{DSACK0}$ and $\overline{DSACK1}$ inputs, as shown in the following table.

Table 13 Effect of $\overline{\text{DSACK}}$ Signals

$\overline{\text{DSACK1}}$	$\overline{\text{DSACK0}}$	Result
1	1	Insert Wait States in Current Bus Cycle
1	0	Complete Cycle — Data Bus Port Size is 8 Bits
0	1	Complete Cycle — Data Bus Port Size is 16 Bits
0	0	Reserved

For example, if the MCU is executing an instruction that reads a long-word operand from a 16-bit port, the MCU latches the 16 bits of valid data and then runs another bus cycle to obtain the other 16 bits. The operation for an 8-bit port is similar, but requires four read cycles. The addressed device uses the $\overline{\text{DSACK0}}$ and $\overline{\text{DSACK1}}$ signals to indicate the port width. For instance, a 16-bit device always returns $\overline{\text{DSACK0}}$ = 1 and $\overline{\text{DSACK1}}$ = 0 for a 16-bit port, regardless of whether the bus cycle is a byte or word operation.

Dynamic bus sizing requires that the portion of the data bus used for a transfer to or from a particular port size be fixed. A 16-bit port must reside on data bus bits [15:0] and an 8-bit port must reside on data bus bits [15:8]. This minimizes the number of bus cycles needed to transfer data and ensures that the MCU transfers valid data.

The MCU always attempts to transfer the maximum amount of data on all bus cycles. For a word operation, it is assumed that the port is 16 bits wide when the bus cycle begins. Operand bytes are designated as shown in the following figure. 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.

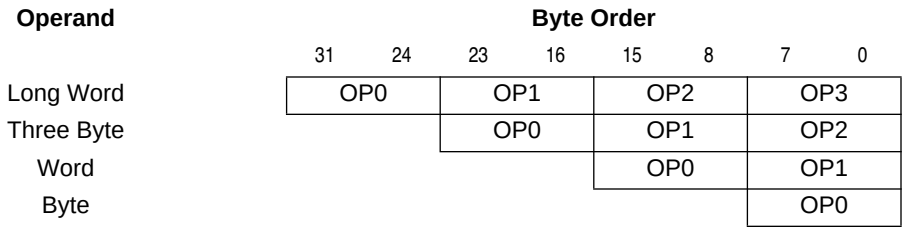


Figure 11 Operand Byte Order

3.5.10 Operand Alignment

The data multiplexer establishes the necessary connections for different combinations of address and data sizes. The multiplexer takes the two bytes of the 16-bit bus and routes them to their required positions. Positioning of bytes is determined by the size and address outputs. SIZ1 and SIZ0 indicate the remaining number of bytes to be transferred during the current bus cycle. The number of bytes transferred is equal to or less than the size indicated by SIZ1 and SIZ0, depending on port width.

ADDR0 also affects the operation of the data multiplexer. During an operand transfer, ADDR[23:1] indicate the word base address of the portion of the operand to be accessed, and ADDR0 indicates the byte offset from the base. Bear in mind the fact that ADDR[23:20] follow the state of ADDR19 in the MC68HC16Z1.

3.5.11 Misaligned Operands

CPU16 processor architecture 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.

In the MC68HC16Z1, the largest amount of data that can be transferred by a single bus cycle is an aligned word. If the MCU transfers a long-word operand via a 16-bit port, the most significant operand word is transferred on the first bus cycle and the least significant operand word on a following bus cycle.

The CPU16 can perform misaligned word transfers. This capability makes it software compatible with the MC68HC11 CPU. The CPU16 treats misaligned long-word transfers as two misaligned word transfers.

3.5.12 Operand Transfer Cases

The following table summarizes 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 SIZ1, SIZ0, and ADDR0 for that bus cycle.

Table 14 Operand Alignment

Transfer Case	SIZ1	SIZ0	ADDR0	$\overline{\text{DSACK1}}$	$\overline{\text{DSACK0}}$	DATA [15:8]	DATA [7:0]
Byte to 8-Bit Port	0	1	X	1	0	OP0	(OP0)
Byte to 16-Bit Port (Even)	0	1	0	0	X	OP0	(OP0)
Byte to 16-Bit Port (Odd)	0	1	1	0	X	(OP0)	OP0
Word to 8-Bit Port (Aligned)	1	0	0	1	0	OP0	(OP1)
Word to 8-Bit Port (Misaligned)	1	0	1	1	0	OP0	(OP0)
Word to 16-Bit Port (Aligned)	1	0	0	0	X	OP0	OP1
Word to 16-Bit Port (Misaligned)	1	0	1	0	X	(OP0)	OP0
3 Byte to 8-Bit Port (Aligned)[2]	1	1	0	1	0	OP0	(OP1)
3 Byte to 8-Bit Port (Misaligned)[2]	1	1	1	1	0	OP0	(OP0)
3 Byte to 16-Bit Port (Aligned)[3]	1	1	0	0	X	OP0	OP1
3 Byte to 16-Bit Port (Misaligned)[2]	1	1	1	0	X	(OP0)	OP0
Long Word to 8-Bit Port (Aligned)	0	0	0	1	0	OP0	(OP1)
Long Word to 8-Bit Port (Misaligned)[3]	1	0	1	1	0	OP0	(OP0)
Long Word to 16-Bit Port (Aligned)	0	0	0	0	X	OP0	OP1
Long Word to 16-Bit Port (Misaligned)[3]	1	0	1	0	X	(OP0)	OP0

NOTES:
1. Operands in parentheses are ignored by the CPU16 during read cycles.
2. Three-byte transfer cases occur only as a result of a long word to byte transfer.
3. The CPU16 treats misaligned long-word transfers as two misaligned word transfers.

3.5.13 Chip Selects

Typical microcontrollers require additional hardware to provide external chip select signals. Twelve independently programmable chip selects provide fast two-cycle access to external memory or peripherals. Address block sizes of two Kbytes to one Mbyte can be selected. However, because ADDR[23:20] = ADDR19 in the CPU16, 512-Kbyte blocks are the largest usable size.

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 $\overline{\text{DSACK}}$ signals internally. A single $\overline{\text{DSACK}}$ generator is shared by all circuits. Multiple chip selects assigned to the same address and control must have the same number of wait states.

Chip selects can also be synchronized with the ECLK signal available on ADDR23.

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 chip select registers. If all parameters match, the appropriate chip select signal is asserted. Select signals are active low. Refer to the following block diagram of a single chip-select circuit.

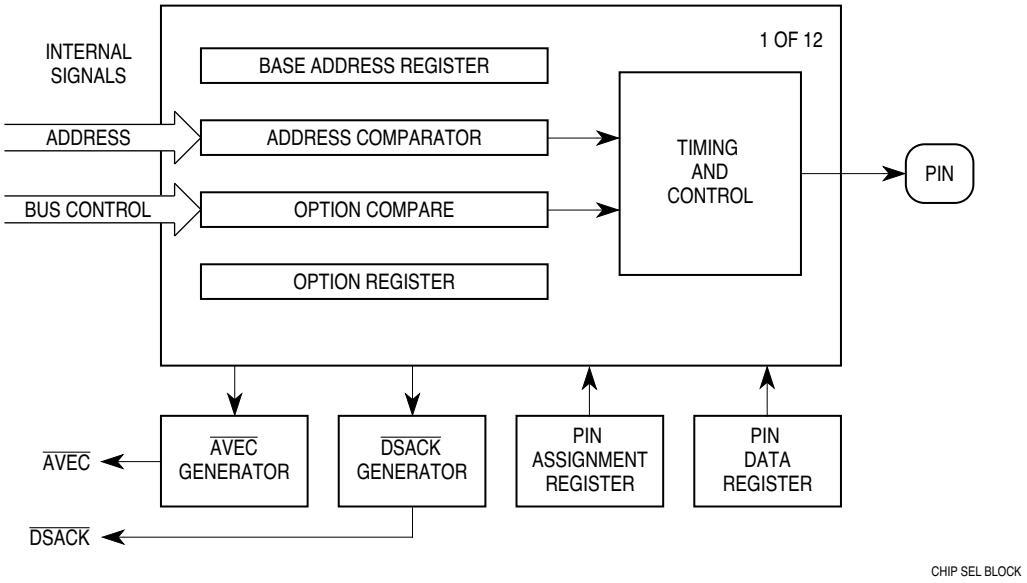


Figure 12 Chip-Select Circuit Block Diagram

Because initialization software usually resides in a peripheral memory device controlled by the chip-select circuits, a $\overline{\text{CSBOOT}}$ register provides default reset values to support bootstrap operation.

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

Each chip select pin can have two or more functions. Chip select configuration out of reset is determined by operating mode. In all modes, the boot ROM select signal is automatically asserted out of reset. In single-chip mode, all chip select pins except $\overline{\text{CS10}}$ and $\overline{\text{CS0}}$ are configured for alternate functions or discrete output. In expanded modes, appropriate pins are configured for chip select operation, but chip select signals cannot be asserted until a transfer size is chosen. In fully expanded mode, data bus pins can be held low to enable alternate functions for chip select pins.

The following table lists allocation of chip-selects and discrete outputs on the pins of the MCU.

Pin	Chip Select	Discrete Outputs
$\overline{\text{CSBOOT}}$	$\overline{\text{CSBOOT}}$	—
$\overline{\text{BR}}$	$\overline{\text{CS0}}$	—
$\overline{\text{BG}}$	$\overline{\text{CS1}}$	—
$\overline{\text{BGACK}}$	$\overline{\text{CS2}}$	—
FC0	$\overline{\text{CS3}}$	PC0
FC1	$\overline{\text{CS4}}$	PC1
FC2	$\overline{\text{CS5}}$	PC2
ADDR19	$\overline{\text{CS6}}$	PC3
ADDR20	$\overline{\text{CS7}}$	PC4
ADDR21	$\overline{\text{CS8}}$	PC5
ADDR22	$\overline{\text{CS9}}$	PC6
ADDR23	$\overline{\text{CS10}}$	ECLK

3.5.13.1 Chip-Select Registers

Pin assignment registers (CSPAR) determine functions of chip select pins. Pin assignment registers also determine port size (8- or 16-bit) for dynamic bus allocation.

A pin data register (PORTC) latches discrete output data.

Blocks of addresses are assigned to each chip select function. Block sizes of 2 Kbytes to 1 Mbyte can be selected by writing values to the appropriate base address register (CSBAR). However, because the logic state of ADDR20 is always the same as the state of ADDR19 in the MC68HC16Z1, the largest usable block size is 512 Kbytes. Address blocks for separate chip select functions can overlap.

Chip select option registers (CSOR) determine timing of and conditions for assertion of chip select signals. Eight parameters, including operating mode, access size, synchronization, and wait state insertion can be specified.

Initialization code often resides in a peripheral memory device controlled by the chip select circuits. A set of special chip select functions and registers (CSORBT, CSBARBT) is provided to support bootstrap operation.

3.5.13.2 Pin Assignment Registers

The pin assignment registers contain pairs of bits that determine the function of pins in other chip-select registers. Alternate functions of the associated pins are shown in parentheses.

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

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

Bits [15:14] — Not Used

These bits always read zero; write has no effect.

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	CS10 (ADDR23)		CS9 (ADDR22)		CS8 (ADDR21)		CS7 (ADDR20)		CS6 (ADDR19)	
RESET:															
0	0	0	0	0	0	DB7	1	DB6	1	DB5	1	DB4	1	DB3	1

Bits [15:10] — Not Used

These bits always read zero; write has no effect.

The following table shows pin assignment register encoding.

Bit Pair	Description
00	Discrete Output
01	Default Function
10	Chip Select (8-Bit Port)
11	Chip Select (16-Bit Port)

A pin programmed as a discrete output drives an external signal to the value specified in the port C data register (PORTC), with the following exceptions:

1. No discrete output function is available on pins $\overline{BR}$, $\overline{BG}$, or $\overline{BGACK}$.
2. ADDR23 provides the ECLK output rather than a discrete output signal.

When a pin is programmed for discrete output or default function, internal chip-select logic still functions and can be used to generate $\overline{\text{DSACK}}$ or $\overline{\text{AVEC}}$ internally on an address match.

Port size is determined when a pin is assigned as a chip select. When a pin is assigned to an 8-bit port, the chip select is asserted at all addresses within the block range. If a pin is assigned to a 16-bit port, the upper/lower byte field of the option register selects the byte with which the chip select is associated.

The notation DB# in a CSPAR reset block indicates that a bit goes to the logic level of that data bus pin on reset. Either default function (01) or chip-select function (11) can be encoded. Because of internal pull-up, DB pins are driven to logic level one by a weak pull-up during reset. Encoding is for chip-select function unless a data line is held low during reset. Note that bus loading can overcome the weak pull-up, and hold pins low during reset. Because ADDR[23:20] follow the state of ADDR19 in the CPU16, DB[7:4] have limited use.

3.5.13.3 Base Address Registers

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

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

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		
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1

CSBAR[10:0] — Chip-Select Base Address Registers **$YFFA4C–$YFFA74**

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		
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 MC68HC16Z1. ADDR[23:20] must match ADDR19 for the chip select to be active.

BLKSZ — Block Size Field

This field determines the size of the block that must be enabled by the chip select. The following table shows bit encoding for the base address registers block size field.

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

ADDR[23:20] is at the same logic level as ADDR19 during normal operation.

ADDR[15:3] — 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. Base address register diagrams show how base register bits correspond to address lines.

Because ADDR20 = ADDR19 in the CPU16, maximum block size is 512 Kbytes. Because ADDR[23:20] follow the logic state of ADDR19, addresses from $080000 to $F7FFFF are inaccessible. Blocks can be based above this dead zone, but the effect of ADDR19 must be considered.

3.5.13.4 Option Registers

The option registers contain eight fields that determine timing of and conditions for assertion of chip-select signals and make the chip selects useful for generating peripheral control signals. All bits in the base address register and the option register must be satisfied to assert a chip-select signal. The bits must also be satisfied to provide $\overline{\text{DSACK}}$ or autovector support.

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

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

CSOR[10:0] — Chip-Select Option Registers **$YFFA4E–$YFFA76**

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

The option register for $\overline{\text{CSBOOT}}$, which is CSORBT, contains special reset values that support bootstrap operations from peripheral memory devices.

The following bit descriptions apply to both CSORBT and CSOR[10:0] option registers.

MODE — Asynchronous/Synchronous Mode

0 = Asynchronous mode selected (chip select assertion determined by internal or external bus control signals)

1 = Synchronous mode selected (chip select assertion synchronized with ECLK signal)

In asynchronous mode, the chip select is asserted synchronized with $\overline{\text{AS}}$ or $\overline{\text{DS}}$.

BYTE — Upper/Lower Byte Option

This field is used only when the chip-select 16-bit port option is selected in the pin assignment register. The following table lists upper/lower byte options.

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

If an interrupting device does not provide a vector number, an autovector acknowledge must be generated. The bus cycle is terminated by asserting $\overline{\text{AVEC}}$. This can be done either by asserting the $\overline{\text{AVEC}}$ pin or by generating $\overline{\text{AVEC}}$ internally, using the chip select option register.

R/$\overline{W}$ — Read/Write

This field causes a chip select to be asserted only for a read, only for a write, or for both read and write. Refer to the following table for options available.

R/$\overline{W}$	Description
00	Reserved
01	Read Only
10	Write Only
11	Read/Write

STRB — Address Strobe/Data Strobe

1 = Data strobe

0 = Address 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 address strobe. Selecting data strobe causes chip select to be asserted synchronized with data strobe.

$\overline{DSACK}$ — Data Strobe Acknowledge

This field specifies the source of $\overline{DSACK}$ in asynchronous mode. It also allows the user to adjust bus timing with internal $\overline{DSACK}$ generation by controlling the number of wait states that are inserted to optimize bus speed in a particular application. The following table shows the $\overline{DSACK}$ field encoding. The fast termination encoding (1110) is used for two-cycle access to external memory.

The $\overline{DSACK}$ field is not used in synchronous mode because a bus cycle is only performed as a synchronous 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.

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

SPACE — Address Space

Use this option field to select an address space for the chip-select logic. The CPU16 normally operates in supervisor space, but interrupt acknowledge must take place in CPU space.

Space Field	Address Space
00	CPU Space
01	User Space
10	Supervisor Space
11	Supervisor/User Space

IPL — Interrupt Priority Level

If the space field is set for CPU space (00), chip-select logic can be used for interrupt acknowledge. During an IACK cycle, the priority level on address lines ADDR[3:1] is compared to the value in the IPL field. If the values are the same, a chip select can be asserted, provided that other option register conditions are met. The following table shows IPL field encoding.

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

This field only affects the response of chip selects and does not affect interrupt recognition by the CPU. Any level means that chip select is asserted regardless of the level of the IACK cycle.

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

1 = Autovector enabled

0 = External interrupt vector enabled

This field selects one of two methods of acquiring an interrupt vector during the IACK cycle. It is not usually used in conjunction with a chip-select pin. If the chip select is configured to trigger on an IACK cycle (SPACE = 00) and the $\overline{\text{AVEC}}$ field is set to one, the chip select automatically generates an $\overline{\text{AVEC}}$ in response to the IACK cycle. Otherwise, the vector must be supplied by the requesting device.

The $\overline{\text{AVEC}}$ bit must not be used in synchronous mode, as autovector response timing can vary because of ECLK synchronization.

PORTC — Port C Data Register **$YFFA41**

7	6	5	4	3	2	1	0
0	PC6	PC5	PC4	PC3	PC2	PC1	PC0
RESET:							
0	1	1	1	1	1	1	1

The data register controls the state of pins programmed as discrete outputs. When a pin is assigned as a discrete output, the value in this register appears at the output. PC[6:0] correspond to CS[9:3]. This is a read/write register. Bit 7 is not used. Writing to this bit has no effect; it always reads zero.

3.5.14 General-Purpose Input/Output

SIM pins can be configured as two general-purpose I/O ports, E and F. The following paragraphs describe registers that control the ports.

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

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

A write to the port E data register is stored in the internal data latch and, if any port E pin is configured as an output, the value stored for that bit is driven on the pin. A read of the port E data register (PORTE) returns the value at the pin only if the pin is configured as a discrete input. Otherwise, the value read is

the value stored in the register. Port E is a single register that can be accessed in two locations. It can be read or written at any time.

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

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

The bits in this register control the direction of the pin drivers when the pins are configured as I/O. Any bit in this register set to one configures the corresponding pin as an output. Any bit in this register cleared to zero configures the corresponding pin as an input. This register can be read or written at any time.

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

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

The bits in this register control the function of each port E pin. Any bit set to one defines the corresponding pin as a bus control signal, with the function shown in the register diagram. Any bit cleared to zero defines the corresponding pin as an I/O pin, controlled by PORTE and DDRE.

Data bus bit 8 controls the state of this register following reset. If DB8 is set to one during reset, the register is set to $FF, which defines all port E pins as bus control signals. If DB8 is cleared to zero during reset, this register is set to $00, defining all port E pins as I/O pins.

NOTE

PE3 is not connected to a pin. PEPA3 returns 1 when read; DDE3 and PE3 bits can be read and written, but have no function.

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

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

The write to the port F data register is stored in the internal data latch, and if any port F pin is configured as an output, the value stored for that bit is driven on the pin. A read of the port F data register (PORTF) returns the value at the pin only if the pin is configured as a discrete input. Otherwise, the value read is the value stored in the register. Port F is a single register that can be accessed in two locations. It can be read or written at any time.

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

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

The bits in this register control the direction of the pin drivers when the pins are configured as I/O. Any bit in this register set to one configures the corresponding pin as an output. Any bit in this register cleared to zero configures the corresponding pin as an input.

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

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

The bits in this register control the function of each port F pin. Any bit set to one defines the corresponding pin to be an interrupt request input as defined in the register diagram. Any bit cleared to zero defines the corresponding pin as an I/O pin, controlled by the port F data and data direction registers. The MODCLK signal has no function after reset.

Data bus bit 9 controls the state of this register following reset. If DB9 is set to one during reset, the register is set to $FF, which defines all port F pins as interrupt request inputs. If DB9 is cleared to zero during reset, this register is set to $00, defining all port F pins as I/O pins.

3.6 Resets

Reset procedures handle system initialization and recovery from catastrophic failure. The MC68HC16Z1 performs resets with a combination of hardware and software. The system integration module 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 SIM. 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 in order 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**

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

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. When multiple reset sources occur at the same time, more than one bit in RSR can be set. 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 system protection submodule 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. Because the CPU16 has no RESET instruction, this bit is not used on the MC68HC16Z1 and always reads zero.

TST — Test Submodule Reset

Reset was caused by the test submodule.

3.6.1 Reset Mode Selection

The logic states of certain data bus pins during reset determine SIM operating configuration. In addition, the state of the MODCLK pin determines system clock source and the state of the pin determines what happens during subsequent breakpoint assertions. The following table is a summary of reset mode selection options.

Table 15 Reset Mode Selection

Mode Select Pin	Default Function (Pin Left High)	Alternate Function (Pin Pulled Low)
DATA0	$\overline{\text{CSBOOT}}$ 16-Bit	$\overline{\text{CSBOOT}}$ 8-Bit
DATA1	$\overline{\text{CS0}}$ $\overline{\text{CS1}}$ $\overline{\text{CS2}}$	$\overline{\text{BR}}$ $\overline{\text{BG}}$ $\overline{\text{BGACK}}$
DATA2	$\overline{\text{CS3}}$ $\overline{\text{CS4}}$ $\overline{\text{CS5}}$	FC0 FC1 FC2
DATA3 DATA4 DATA5 DATA6 DATA7	$\overline{\text{CS6}}$ $\overline{\text{CS7}}$–$\overline{\text{CS6}}$ $\overline{\text{CS8}}$–$\overline{\text{CS6}}$ $\overline{\text{CS9}}$–$\overline{\text{CS6}}$ $\overline{\text{CS10}}$–$\overline{\text{CS6}}$	ADDR19 ADDR[20:19] ADDR[21:19] ADDR[22:19] ADDR[23:19]
DATA8	$\overline{\text{DSACK0}}$, $\overline{\text{DSACK1}}$, $\overline{\text{AVEC}}$, $\overline{\text{DS}}$, $\overline{\text{AS}}$, $\overline{\text{SIZE}}$	PORTE
DATA9	$\overline{\text{IRQ7}}$–$\overline{\text{IRQ1}}$ MODCLK	PORTF
DATA11	Test Mode Disabled	Test Mode Enabled
DATA14	ROM STOP = 0 (Enabled)	ROM STOP = 1 (Disabled)
MODCLK	VCO = System Clock	EXTAL = System Clock
$\overline{\text{BKPT}}$	Background Mode Disabled	Background Mode Enabled

3.6.2 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. The following table is summary of module pin function out of reset.

Table 16 Module Pin Functions

Module	Pin Mnemonic	Function
ADC	PADA[7:0]/AN[7:0]	DISCRETE INPUT
	V_{RH}	REFERENCE VOLTAGE
	V_{RL}	REFERENCE VOLTAGE
CPU	DSI/IPIPE1	DSI/IPIPE1
	DSO/IPIPE0	DSO/IPIPE0
	$\overline{BKPT}$/DSCLK	$\overline{BKPT}$/DSCLK
GPT	PGP7/IC4/OC5	DISCRETE INPUT
	PGP[6:3]/OC[4:1]	DISCRETE INPUT
	PGP[2:0]/IC[3:1]	DISCRETE INPUT
	PAI	DISCRETE INPUT
	PCLK	DISCRETE INPUT
	PWMA, PWMB	DISCRETE OUTPUT
QSM	PQS7/TXD	DISCRETE INPUT
	PQS[6:4]/PCS[3:1]	DISCRETE INPUT
	PQS3/PCS0/$\overline{SS}$	DISCRETE INPUT
	PQS2/SCK	DISCRETE INPUT
	PQS1/MOSI	DISCRETE INPUT
	PQS0/MISO	DISCRETE INPUT
	RXD	RXD

3.6.3 Reset Timing

The $\overline{RESET}$ input must be asserted for a specified minimum period in order for reset to occur. External $\overline{RESET}$ 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 $\overline{RESET}$ is asserted, SIM pins are either in an inactive, high impedance state or are driven to their inactive states.

When an external device asserts $\overline{RESET}$ for the proper period, reset control logic clocks the signal into an internal latch. The control logic drives the $\overline{RESET}$ pin low for an additional 512 CLKOUT cycles after it detects that the $\overline{RESET}$ 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 $\overline{RESET}$ 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 $\overline{RESET}$ 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 reset input is tested. When the input is at logic level one, reset exception processing begins. If, however, the 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 $\overline{RESET}$ is released.

3.6.4 Power-On Reset

When the SIM clock synthesizer is used to generate system clocks, power-on reset involves special circumstances related to application of system and clock synthesizer power. Regardless of clock source, voltage must be applied to clock synthesizer power input pin V_{DDSYN}, in order for the MCU to operate. The following discussion assumes that V_{DDSYN} is applied before and during reset — this minimizes crystal start-up time. 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 SIM drives the IMB internal and external reset lines. The circuit releases the internal reset line as V_{DD} ramps up to the minimum specified value, and SIM 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 SIM clock synthesizer provides clock signals to the other MCU modules. After the clock is running and the internal 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.

3.6.4.1 Use of Three State Control Pin

Asserting the three-state control (TSC) input causes the MCU to put all output drivers in an inactive, high-impedance state. The signal must remain asserted for 10 clock cycles in order for drivers to change state. There are certain constraints on use of TSC during power-up reset:

When the internal clock synthesizer is used (MODCLK held high during reset), synthesizer ramp-up time affects how long the 10 cycles take. Worst case is approximately 20 milliseconds from TSC assertion.

When an external clock signal is applied (MODCLK held low during reset), pins go to high-impedance state as soon after TSC assertion as 10 clock pulses have been applied to the EXTAL pin.

When 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 system integration module, and a device or module requesting interrupt service.

The CPU16 provides for eight levels of interrupt priority (0–7), 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 interrupt request signals $\overline{\text{IRQ}}$[7:1] and the IP mask value. Each of the signals corresponds to an interrupt priority. $\overline{\text{IRQ1}}$ has the lowest priority, and $\overline{\text{IRQ7}}$ has the highest priority.

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 $\overline{\text{IRQ7}}$) 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 SIM interrupt control logic — the CPU treats external interrupt requests as though they come from the SIM.

External $\overline{IRQ}$[6:1] are active-low level-sensitive inputs. External is an active-low transition-sensitive input — it requires both an edge and a voltage level for validity.

$\overline{IRQ}$[6:1] are maskable.$\overline{IRQ7}$ is nonmaskable. The $\overline{IRQ7}$ input is transition-sensitive in order to prevent redundant servicing and stack overflow. A nonmaskable 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.

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 and Arbitration

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 bit values.

Each module that can make an interrupt service request, including the SIM, has an IARB field in its 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 SIM IARB value is used for arbitration between internal and external interrupt requests. The reset value of IARB for the SIM is %1111, and the reset IARB value for all other modules is %0000. Initialization software must assign different IARB values in order to implement an arbitration scheme.

Each module must have a unique IARB value. When two or more IARB fields have the same nonzero 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 SIM 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 ac-

knowledge 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.

The periodic interrupt timer (PIT) in the SIM can generate internal interrupt requests of specific priority at predetermined intervals. By hardware convention, PIT interrupts are serviced before external interrupt service requests of the same priority. Refer to **3.4.4 Periodic Interrupt Timer** for more information.

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.

- The CPU finishes higher priority exception processing or reaches an instruction boundary.
- Processor state is stacked, then the CCR PK extension field is cleared.
- The interrupt acknowledge cycle begins:
  - — FC[2:0] are driven to %111 (CPU space) encoding.
  - — 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] = %11111111111; ADDR[3:1] = the priority of the interrupt request being acknowledged; and ADDR0 = %1.
  - — Request priority is latched into the CCR IP field from the address bus.
- Modules or external peripherals that have requested interrupt service decode the priority value in 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, and a spurious interrupt exception is processed.
- After arbitration, the interrupt acknowledge cycle can be completed in one of three ways:
  - — The dominant interrupt source supplies a vector number and signals appropriate to the access. The CPU16 acquires the vector number.
  - — The signal is asserted (the signal can be asserted by the dominant interrupt source or the pin can be tied low), and the CPU16 generates an autovector number corresponding to interrupt priority.
  - — The bus monitor asserts and the CPU16 generates the spurious interrupt vector number.
- The vector number is converted to a vector address.
- The content of the vector address is loaded into the PC, and the processor transfers control to the exception handler routine.

3.8 Factory Test Block

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

3.8.1 Test Registers

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

SIMTR — System Integration Test Register **$YFFA02**

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

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

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

TSTSC — Test Module Shift Count **$YFFA34**

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

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

DREG — Distributed Register **$YFFA3A**

4 Analog-to-Digital Converter Module

The ADC is a unipolar, successive-approximation converter with eight modes of operation. It has selectable 8- or 10-bit resolution. Accuracy is ±1 count (1 LSB) in 8-bit mode and ± 2.5 counts (2.5 LSB) in 10-bit mode. Monotonicity is guaranteed in both modes. With a 16.78-MHz clock, the ADC can perform an 8-bit single conversion (4-clock sample) in 8 microseconds, a 10-bit single conversion in 9 microseconds.

ADC functions can be grouped into three subsystems: an analog front end, a digital control section, and a bus interface. A block diagram of the converter appears on the following page.

4.1 Analog Subsystem

The analog front end consists of a multiplexer, input sample buffer amplifier, a resistor-capacitor array, and a high-gain comparator. The multiplexer selects one of eight internal or eight external signal sources for conversion. The resistor capacitor (RC) array performs two functions. It acts as a sample/hold circuit, and it provides the digital-to-analog comparison output necessary for successive approximation conversion. The comparator indicates whether each successive output of the RC array is higher or lower than the sampled input.

4.2 Digital Control Subsystem

The digital control section includes conversion sequence control logic, channel and reference select logic, successive approximation register, eight result registers, a port data register, and control/status registers. It controls the multiplexer and the output of the RC array during the sample and conversion periods, stores the results of comparison in the successive-approximation register, then transfers the result to a result register.

4.3 Bus Interface Subsystem

The bus interface contains logic necessary to interface the ADC to the intermodule bus. The ADC is designed to act as a slave device on the bus. The interface must respond with appropriate bus cycle termination signals and must supply appropriate interface timing to the other submodule.

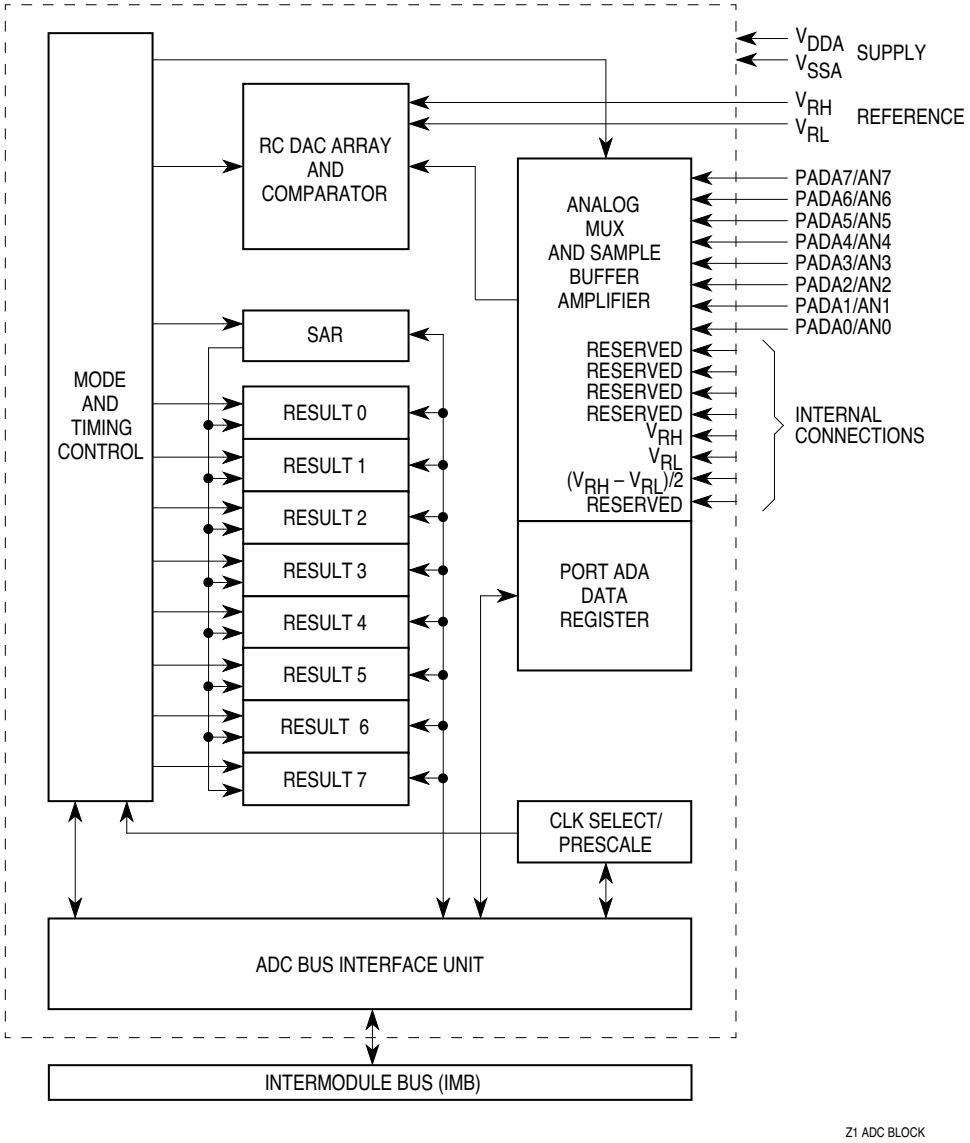


Figure 13 Analog-to-Digital Converter Block Diagram

Table 17 ADC Address Map

Address	15 8	7 0
$YFF700	MODULE CONFIGURATION (ADCMCR)	
$YFF702	FACTORY TEST (ADTEST)	
$YFF704	(RESERVED)	
$YFF706	PORT ADA DATA (PORTADA)	
$YFF708	(RESERVED)	
$YFF70A	ADC CONTROL 0 (ADCTL0)	
$YFF70C	ADC CONTROL 1 (ADCTL1)	
$YFF70E	ADC STATUS (ADSTAT)	
$YFF710	RIGHT-JUSTIFIED UNSIGNED RESULT 0 (RJURR0)	
$YFF712	RIGHT-JUSTIFIED UNSIGNED RESULT 1 (RJURR1)	
$YFF714	RIGHT-JUSTIFIED UNSIGNED RESULT 2 (RJURR2)	
$YFF716	RIGHT-JUSTIFIED UNSIGNED RESULT 3 (RJURR3)	
$YFF718	RIGHT-JUSTIFIED UNSIGNED RESULT 4 (RJURR4)	
$YFF71A	RIGHT-JUSTIFIED UNSIGNED RESULT 5 (RJURR5)	
$YFF71C	RIGHT-JUSTIFIED UNSIGNED RESULT 6 (RJURR6)	
$YFF71E	RIGHT-JUSTIFIED UNSIGNED RESULT 7 (RJURR7)	
$YFF720	LEFT-JUSTIFIED SIGNED RESULT 0 (LJSRR0)	
$YFF722	LEFT-JUSTIFIED SIGNED RESULT 1 (LJSRR1)	
$YFF724	LEFT-JUSTIFIED SIGNED RESULT 2 (LJSRR2)	
$YFF726	LEFT-JUSTIFIED SIGNED RESULT 3 (LJSRR3)	
$YFF728	LEFT-JUSTIFIED SIGNED RESULT 4 (LJSRR4)	
$YFF72A	LEFT-JUSTIFIED SIGNED RESULT 5 (LJSRR5)	
$YFF72C	LEFT-JUSTIFIED SIGNED RESULT 6 (LJSRR6)	
$YFF72E	LEFT-JUSTIFIED SIGNED RESULT 7 (LJSRR7)	
$YFF730	LEFT-JUSTIFIED UNSIGNED RESULT 0 (LJURR0)	
$YFF732	LEFT-JUSTIFIED UNSIGNED RESULT 1 (LJURR1)	
$YFF734	LEFT-JUSTIFIED UNSIGNED RESULT 2 (LJURR2)	
$YFF736	LEFT-JUSTIFIED UNSIGNED RESULT 3 (LJURR3)	
$YFF738	LEFT-JUSTIFIED UNSIGNED RESULT 4 (LJURR4)	
$YFF73A	LEFT-JUSTIFIED UNSIGNED RESULT 5 (LJURR5)	
$YFF73C	LEFT-JUSTIFIED UNSIGNED RESULT 6 (LJURR6)	
$YFF73E	LEFT-JUSTIFIED UNSIGNED RESULT 7 (LJURR7)	

Y = M111, where M is the logic state of the modmap (MM) bit in the SIMCR

4.4 ADC Registers

ADCMCR — ADC Module Configuration Register **$YFF700**

15	14	13	12 8	7	6 0
STOP	FRZ		NOT USED	SUPV	NOT USED
RESET:					
1	0	0		0	

Use the module configuration register to initialize the ADC.

STOP — STOP Mode

0 = Normal operation

1 = Low-power operation

STOP places the ADC in low-power state by disabling the ADC clock and powering down the analog circuitry. Setting STOP aborts any conversion in progress. STOP is set to logic level one at reset and can be cleared to logic level zero by the CPU.

Clearing STOP enables normal ADC operation. However, because analog circuitry bias current has been turned off, there is a period of recovery before output stabilization.

FRZ[1:0] — Freeze 1

Use the FRZ field to determine ADC response to assertion of the IFREEZE signal. The following table shows possible responses.

FRZ	Response
00	Ignore IFREEZE
01	Reserved
10	Finish conversion, then freeze
11	Freeze immediately

SUPV — Supervisor/Unrestricted

0 = Unrestricted access

1 = Supervisor access

SUPV defines access to assignable ADC registers. Because the CPU16 in the MC68HC16Z1 operates in supervisor mode only, this bit has no effect.

ADTEST — ADC Test Register **$YFF702**

ADTEST is used with the SIM test register for factory test of the ADC.

PORTADA — Port ADA Data Register **$YFF706**

15	11	10	9	8	7	0
NOT USED					PORTADA	
RESET:						
0	0	0	0	0	0	0	0	INPUT DATA

Port A is an input port that shares pins with the A/D converter inputs.

PORTADA [7:0]

A read of PORTADA[7:0] returns the logic level of the port A pins. If the input is not an appropriate voltage (outside the defined levels), the read will be indeterminate. Use of a port A pin for digital input does not preclude its use as an analog input.

ADCTL0 — A/D Control Register 0 **$YFF70A**

15	8	7	6	5	4	3	2	1	0
NOT USED		RES10	STS		PRS				
RESET:									
0	0	0	0	0	0	0	0	1	1

Use ADCTL0 to select ADC clock source and to set up prescaling. Writes to it have immediate effect.

RES10 — 10-Bit Resolution

0 = 8-bit conversion

1 = 10-bit conversion

Conversion results are appropriately aligned in result registers to reflect conversion status.

STS[1:0] — Sample Time Select Field

Total conversion time depends on initial sample time, transfer time, final sample time, and resolution time. Initial sample time is fixed at two clocks. Transfer time is fixed at two clocks. Resolution time is fixed at 10 ADC clock cycles for an 8-bit conversion and 12 ADC clock cycles for a 10-bit conversion. Final sample time is determined by the value in the STS field, as shown in the following table.

STS[1:0]	Final Sample Time
00	2 A/D Clock Periods
01	4 A/D Clock Periods
10	8 A/D Clock Periods
11	16 A/D Clock Periods

PRS[4:0] — Prescaler Rate Selection Field

ADC clock is generated from system clock using a modulo counter and a divide-by-two circuit. The binary value of this field is the counter modulus. System clock is divided by the PRS value plus one, then sent to the divide-by-two circuit, as shown in the following table.

PRS[4:0]	Divisor Value	Max. System Clock	Min. System Clock
00000	RESERVED	—	—
00001	4	8 MHz	2 MHz
00010	6	12 MHz	3 MHz
...	...	...	...
11101	60	120 MHz	30 MHz
11110	62	124 MHz	31 MHz
11111	64	128 MHz	32 MHz

ADCTL1 — A/D Control Register 1 **$YFF70C**

15	8	7	6	5	4	3	2	1	0
NOT USED			SCAN	MULT	S8CM	CD	CC	CB	CA
RESET:									
0	0	0	0	0	0	0	0	0	0

Use ADCTL1 to initiate A/D conversion, or to select conversion modes and conversion channel. It can be written or read at any time. A write to ADCTL1 initiates a conversion sequence. If a conversion sequence is already in progress, a write to ADCTL1 aborts it and resets the SCF and CCF flags in the A/D status register.

SCAN — Scan Mode Selection Bit

0 = Single conversion sequence
1 = Continuous conversion

Length of conversion sequence(s) is determined by S8CM.

MULT — Multichannel Conversion Bit

0 = Conversion sequence(s) run on single channel (channel selected by [CD:CA])
1 = Sequential conversion of a block of four or eight channels (block selected by [CD:CA])

Length of conversion sequence(s) is determined by S8CM.

S8CM — Select Eight-Conversion Sequence Mode

0 = Four-conversion sequence
1 = Eight-conversion sequence

This bit determines the number of conversions in a conversion sequence.

[CD:CA] — Channel Selection Field

Use the bits in this field to select an input or block of inputs for A/D conversion.

The following table is a summary of the operation of S8CM and [CD:CA] when MULT is cleared (single-channel mode). Number of conversions per channel is determined by SCAN.

S8CM	CD	CC	CB	CA	Input	Result Register
0	0	0	0	0	AN0	RSLT[0:3]
0	0	0	0	1	AN1	RSLT[0:3]
0	0	0	1	0	AN2	RSLT[0:3]
0	0	0	1	1	AN3	RSLT[0:3]
0	0	1	0	0	AN4	RSLT[0:3]
0	0	1	0	1	AN5	RSLT[0:3]
0	0	1	1	0	AN6	RSLT[0:3]
0	0	1	1	1	AN7	RSLT[0:3]
0	1	0	0	0	RESERVED	RSLT[0:3]
0	1	0	0	1	RESERVED	RSLT[0:3]
0	1	0	1	0	RESERVED	RSLT[0:3]
0	1	0	1	1	RESERVED	RSLT[0:3]
0	1	1	0	0	V_{RH}	RSLT[0:3]
0	1	1	0	1	V_{RL}	RSLT[0:3]
0	1	1	1	0	$(V_{RH} - V_{RL}) / 2$	RSLT[0:3]
0	1	1	1	1	TEST/RESERVED	RSLT[0:3]
1	0	0	0	0	AN0	RSLT[0:7]
1	0	0	0	1	AN1	RSLT[0:7]
1	0	0	1	0	AN2	RSLT[0:7]
1	0	0	1	1	AN3	RSLT[0:7]
1	0	1	0	0	AN4	RSLT[0:7]
1	0	1	0	1	AN5	RSLT[0:7]
1	0	1	1	0	AN6	RSLT[0:7]
1	0	1	1	1	AN7	RSLT[0:7]
1	1	0	0	0	RESERVED	RSLT[0:7]
1	1	0	0	1	RESERVED	RSLT[0:7]
1	1	0	1	0	RESERVED	RSLT[0:7]
1	1	0	1	1	RESERVED	RSLT[0:7]
1	1	1	0	0	V_{RH}	RSLT[0:7]
1	1	1	0	1	V_{RL}	RSLT[0:7]
1	1	1	1	0	$(V_{RH} - V_{RL}) / 2$	RSLT[0:7]
1	1	1	1	1	TEST/RESERVED	RSLT[0:7]

The following table is a summary of the operation of S8CM and [CD:CA] when MULT is set (multi-channel mode). Number of conversions per channel is determined by SCAN. Channel numbers are given in order of conversion.

S8CM	CD	CC	CB	CA	Input	Result Register
0	0	0	X	X	AN0	RSLT0
					AN1	RSLT1
					AN2	RSLT2
					AN3	RSLT3
0	0	1	X	X	AN4	RSLT0
					AN5	RSLT1
					AN6	RSLT2
					AN7	RSLT3
0	1	0	X	X	RESERVED	RSLT0
					RESERVED	RSLT1
					RESERVED	RSLT2
					RESERVED	RSLT3
0	1	1	X	X	V_{RH}	RSLT0
					V_{RL}	RSLT1
					$(V_{RH} - V_{RL}) / 2$	RSLT2
					TEST/RESERVED	RSLT3
1	0	X	X	X	AN0	RSLT0
					AN1	RSLT1
					AN2	RSLT2
					AN3	RSLT3
					AN4	RSLT4
					AN5	RSLT5
					AN6	RSLT6
					AN7	RSLT7
1	1	X	X	X	RESERVED	RSLT0
					RESERVED	RSLT1
					RESERVED	RSLT2
					RESERVED	RSLT3
					V_{RH}	RSLT4
					V_{RL}	RSLT5
					$(V_{RH} - V_{RL}) / 2$	RSLT6
					TEST/RESERVED	RSLT7

ADSTAT — ADC Status Register **$YFF70E**

15	14	11	10	8	7	0
SCF	NOT USED		CCTR		CCF	

RESET:

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ADSTAT contains information related to the status of a conversion sequence.

SCF — Sequence Complete Flag

0 = Sequence not complete

1 = Sequence complete

SCF is set at the end of the conversion sequence when SCAN is cleared, and at the end of the **first** conversion sequence when SCAN is set. SCF is cleared when ADCTL1 is written and a new conversion sequence begins.

CCTR[2:0] — Conversion Counter Field

This field reflects the contents of the conversion counter pointer in either four or eight count conversion sequence. The value corresponds to the number of the next result register to be written, and thus indicates which channel is being converted.

CCF[7:0] — Conversion Complete Field

Each bit in this field corresponds to an A/D result register (CCF7 to RSLT7, etc.). A bit is set when conversion for the corresponding channel is complete, and remains set until the result register is read. A bit is cleared when the register is read.

RSLT[0:7] — A/D Result Registers **$YFF710–$YFF73E**

The result registers store data after conversion is complete. Each register can be read from three different addresses in the register block. Data format depends on the address from which the data is read.

RJURR — Unsigned Right-Justified Format **$YFF710–$YFF71E**

Conversion result is unsigned right-justified data. Bits [9:0] are used for 10-bit resolution, bits [7:0] are used for 8-bit conversion (bits [9:8] are zero). Bits [15:10] always return zero when read.

LJSRR — Signed Left-Justified Format **$YFF720–$YFF72E**

Conversion result is signed left-justified data. Bits [15:6] are used for 10-bit resolution, and bits [15:8] are used for 8-bit conversion (bits [7:6] are zero). Although the ADC is unipolar, it is assumed that the zero point is halfway between low and high reference when this format is used. For positive input, bit 15 = 0. For negative input, bit 15 = 1. Bits [5:0] always return zero when read.

LJURR — Unsigned Left-Justified Format **$YFF730–$YFF73E**

Conversion result is unsigned left-justified data. Bits [15:6] are used for 10-bit resolution, and bits [15:8] are used for 8-bit conversion (bits [7:6] are zero). Bits [5:0] always return zero when read.

5 Queued Serial Module

The QSM contains two serial interfaces, the queued serial peripheral interface (QSPI) and the serial communication interface (SCI).

The QSPI provides easy peripheral expansion or interprocessor communication through a full-duplex, synchronous, three-line bus: data in, data out, and a serial clock. Four programmable peripheral-select pins provide addressability for up to 16 peripheral devices. A self-contained RAM queue allows up to 16 serial transfers of 8–16 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 nonreturn 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 64 to 524 kbaud with a 16.78-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.

Refer to the following block diagram of the QSM.

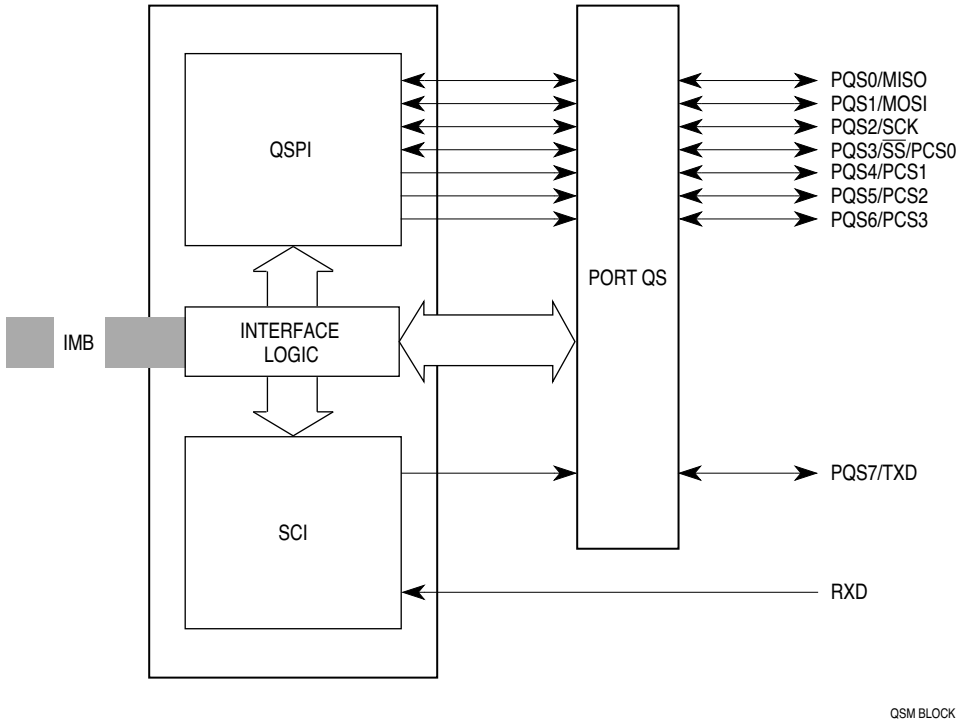


Figure 14 QSM Block Diagram

Table 18 QSM Address Map

Address	15 8	7 0
$YFFC00	QSM MODULE CONFIGURATION (QSMCR)	
$YFFC02	QSM TEST (QTEST)	
$YFFC04	QSM INTERRUPT LEVEL (QILR)	QSM INTERRUPT VECTOR (QIVR)
$YFFC06	RESERVED	
$YFFC08	SCI CONTROL 0 (SCCR0)	
$YFFC0A	SCI CONTROL 1 (SCCR1)	
$YFFC0C	SCI STATUS (SCSR)	
$YFFC0E	SCI DATA (SCDR)	
$YFFC10	RESERVED	
$YFFC12	RESERVED	
$YFFC14	RESERVED	PQS DATA (PORTQS)
$YFFC16	PQS PIN ASSIGNMENT (PQSPAR)	PQS DATA DIRECTION (DDRQS)
$YFFC18	SPI CONTROL 0 (SPCR0)	
$YFFC1A	SPI CONTROL 1 (SPCR1)	
$YFFC1C	SPI CONTROL 2 (SPCR2)	
$YFFC1E	SPI CONTROL 3 (SPCR3)	SPI STATUS (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])	

Y = M111, where M is the logic state of the modmap (MM) bit in the SIMCR

The following table 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 input or output.

	Pin	Mode	Pin Function
QSPI Pins	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
SCI Pins	TXD	Transmit	Serial Data Output from SCI
	RXD	Receive	Serial Data Input to SCI

5.1 QSM Registers

There are four types of QSM registers: 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.

The modmap (MM) bit in the system integration module configuration register (SIMCR) defines the most significant bit (ADDR23) of the address, shown in each register figure as Y. This bit, concatenated with the rest of the address given, forms the absolute address of each register. Because the CPU16 in the MC68HC16Z1 drives only ADDR[19:0], ADDR[23:20] follow the logic state of ADDR19, and Y must equal $F. Refer to the SIM section of this technical summary for more information about how the state of MM affects the system.

5.1.1 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			
RESET:															
0	0	0	0	0	0	0	0	1	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. 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 1

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 — Freeze 0

Reserved

Bits [12:8] — Not Implemented

SUPV — Supervisor/Unrestricted

0 = User access
1 = Supervisor access (MC68HC16Z1 default)

SUPV defines the assignable QSM registers as either supervisor-only data space or unrestricted data space. Because the CPU16 in the MC68HC16Z1 operates in supervisor mode only, this bit has no effect.

Bits [6:4] — Not Implemented

IARB — Interrupt Arbitration Identification Number

Each module that generates interrupts must have an IARB field. In this field, each module has a unique value that is used to arbitrate for the IMB when modules generate simultaneous interrupts of the same priority. Refer to the SIM section of this summary for more information.

QTEST — QSM Test Register **$YFFC02**

QTEST is used during factory test of the QSM. Accesses to QTEST must be made while the MCU is in test mode.

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

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

QILR determines the priority level of interrupts requested by the QSM and the vector used when an interrupt is acknowledged.

ILQSPI — 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 — 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	8	7	6	5	4	3	2	1	0
QILR		INTV							
		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 IACK cycle is supplied by the QSM. During an IACK, 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.

5.1.2 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 input/output (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 **$YFFC15**

15 … 8	7	6	5	4	3	2	1	0
RESERVED	DATA7 (TXD)	DATA6 (PCS3)	DATA5 (PCS2)	DATA4 (PCS1)	DATA3 (PCS0/$\overline{SS}$)	DATA2 (SCK)	DATA1 (MOSI)	DATA0 (MISO)
RESET:	0	0	0	0	0	0	0	0

PORTQS is the port QS data register. Writes to PORTQS affect pins defined as outputs. Reads of PORTQS return data present on the pins.

PQSPAR — Port QS Pin Assignment Register **$YFFC16**

15	14	13	12	11	10	9	8	7 … 0
0	PCS3	PCS2	PCS1	PCS0/$\overline{SS}$	0	MOSI	MISO	DDRQS
RESET:								
0	0	0	0	0	0	0	0	

PQSPAR determines whether certain pins are used by the QSPI submodule, or whether they are available for general-purpose I/O. Pins designated for general-purpose I/O are controlled by DDRQS and PORTQS. PQSPAR does not affect operation of the SCI submodule. Bits 15 and 10 are not implemented.

PCS[3:1] — Peripheral Chip Selects
PCS0/$\overline{SS}$ — Peripheral Chip Select 0/Slave Select
MOSI — Master Out Slave In
MISO — Master In Slave Out
0 = Used for general-purpose I/O
1 = Used by QSPI submodule

DDRQS — Port QS Data Direction Register **$YFFC17**

15 … 8	7	6	5	4	3	2	1	0
PQSPAR	TXD	PCS3	PCS2	PCS1	PCS0/$\overline{SS}$	SCK	MOSI	MISO
RESET:								
	0	0	0	0	0	0	0	0

DDRQS determines whether a general-purpose I/O pin is an input or an output. During reset, all QSM pins are configured as general-purpose inputs.

TXD — Transmit Data
0 = Input
1 = Output
This bit 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.

All of the following bits determine the corresponding QSPI port pin operation to be input or output.

PCS[3:1] — Peripheral Chip Selects
PCS0/$\overline{SS}$ — Peripheral Chip Select 0/Slave Select
SCK — Serial Clock
MOSI — Master Out Slave In
MISO — Master In Slave Out
0 = Input
1 = Output

5.2 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. Refer to the following block diagram of the QSPI.

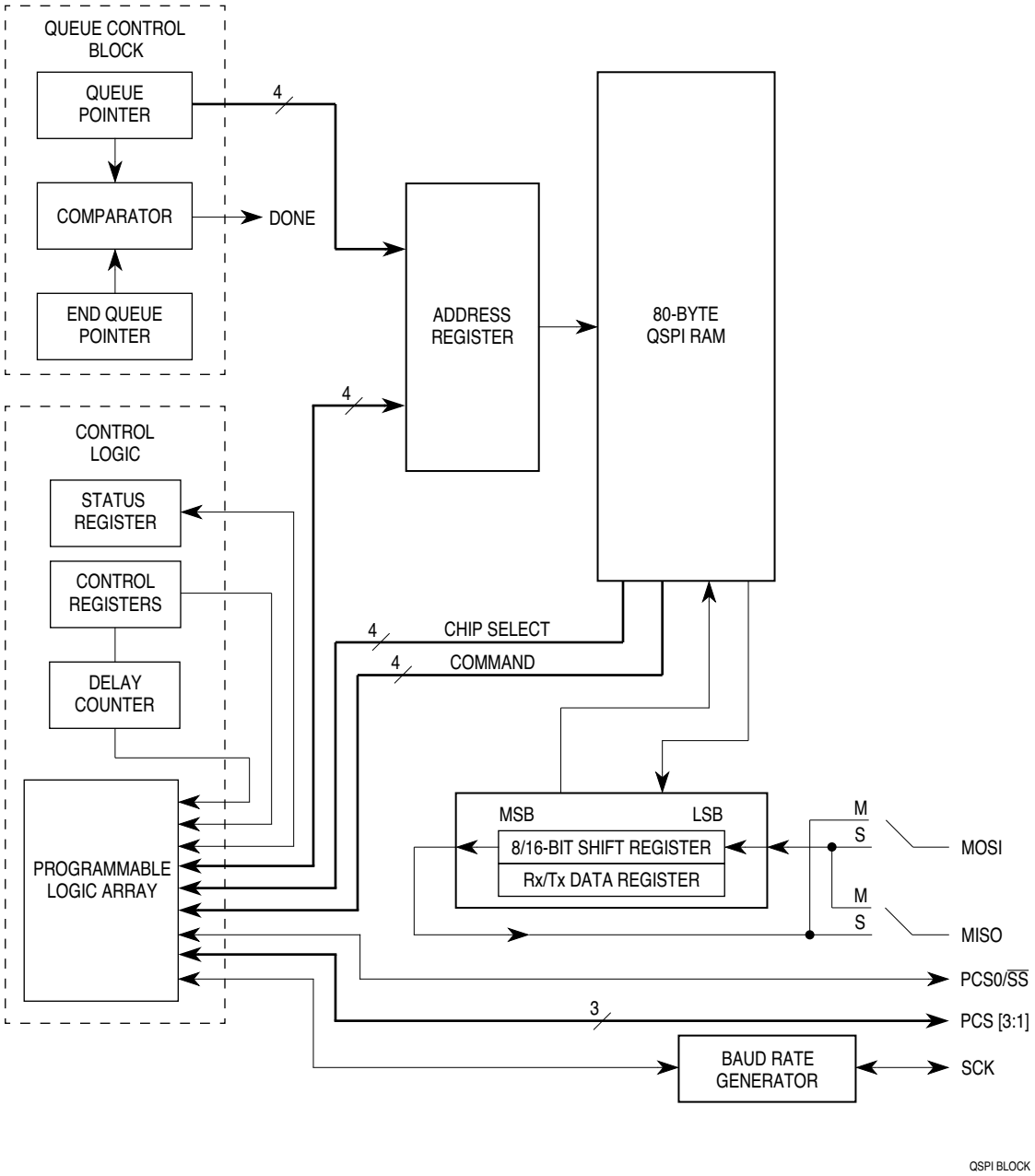


Figure 15 QSPI Block Diagram

5.2.1 QSPI Pins

Seven pins are associated with the QSPI. When not needed for a QSPI application, they can be configured as general-purpose I/O pins.

Refer to the following table for QSPI input and output pins and their functions.

Pin Names	Mnemonics	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[3:0]	Master	Select Peripherals
Slave Select	$\overline{SS}$	Master Slave	Causes Mode Fault Initiates Serial Transfer

5.2.2 QSPI Registers

The programmer's 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.

Registers and RAM can be read and written by the CPU. The four control registers must be initialized before the QSPI is enabled to insure 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.

Refer to the following memory map of the QSPI.

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	RAM	QSPI Receive Data (16 Words)
$YFFD20	RAM	QSPI Transmit Data (16 Words)
$YFFD40	RAM	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 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				CPOL	CPHA	SPBR							
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 whether the QSPI is enabled or disabled.

BITS — Bits Per Transfer

In master mode, when BITSE in a command is set, the BITS field determines the number of data bits transferred. When BITSE is cleared, eight bits are transferred. Reserved values default to eight bits. BITSE is not used in slave mode.

The following table shows the number of bits per transfer.

BITS	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 — 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 field. The following equation determines the SCK baud rate:

SCK Baud Rate = System Clock/(2SPBR)

or

SPBR = System Clock/(2SCK)(Baud Rate Desired)

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

Giving SPBR 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 is initialized to a 2.1-MHz SCK frequency.

SPCR1 — QSPI Control Register 1 **$YFFC1A**

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

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 — 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:

PCS to SCK Delay = [DSCKL/System Clock]

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

When a queue entry's DSCK equals zero, then DSCKL is not used. Instead, the PCS valid-to-SCK transition is one-half SCK period.

DTL — 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:

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

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

A zero value for DTL causes a delay-after-transfer value of 8192/System Clock.

If DT equals zero, a standard delay is inserted.

Standard Delay after Transfer = [17/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				0	0	0	0	NEWQP			
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SPCR2 contains QSPI configuration parameters. Although 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 — Ending Queue Pointer
This field contains the last QSPI queue address.

Bits [7:4] — Not Implemented

NEWQP — New Queue Pointer Value
This field contains the first QSPI queue address.

SPCR3 — QSPI Control Register 3 **$YFFC1E**

15	14	13	12	11	10	9	8	7	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 … 8	7	6	5	4	3	2	1	0
SPCR3	SPIF	MODF	HALTA	0	CPTQP			
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.

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).

MODF is asserted by the QSPI 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 — Completed Queue Pointer

CPTQP 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 contains either the reset value ($0) or a pointer to the last command completed in the previous queue.

5.2.3 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 RAM, transmit data RAM, and command control RAM. 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.

Refer to the following illustration of the organization of the RAM.

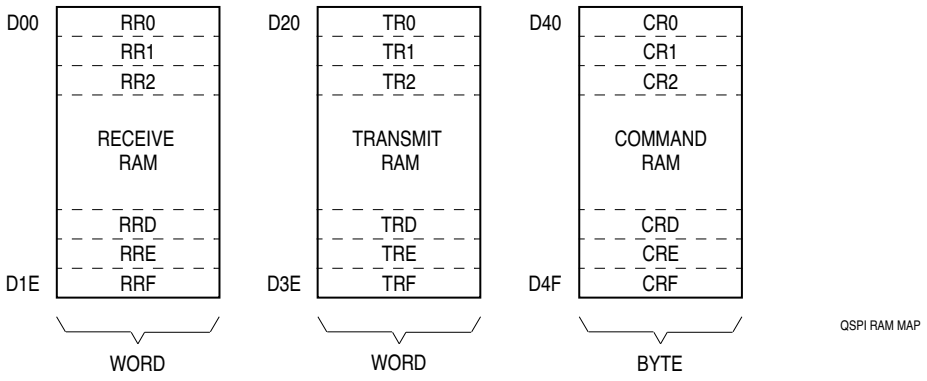


Figure 16 QSPI RAM Address Map

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 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*
–	–	–	–	–	–	–	–
CONT	BITSE	DT	DSCK	PCS3	PCS2	PCS1	PCS0*

COMMAND CONTROL | PERIPHERAL CHIP SELECT

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

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

maximum of 16 commands can be in the queue. Queue execution by the QSPI proceeds from the address in NEWQP through the address in ENDQP (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 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 field.

DSCK — PCS to SCK Delay

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

PCS[3:0] — Peripheral Chip Select

Use peripheral chip-select bits to select an external 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.

$\overline{SS}$ — Slave Mode Select

Initiates slave mode serial transfer. If $\overline{SS}$ is taken low when the QSPI is in master mode, a mode fault will be generated.

5.2.4 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 data RAM and received into receive data 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 multimaster 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.

5.3 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.

5.3.1 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.

The following table shows SCI pins and their functions.

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

5.3.2 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 register 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												
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 — Baud Rate

SCI baud rate is programmed by writing a 13-bit value to SCBR. 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:

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

or

$$\text{SCBR} = \text{System Clock}(32\text{SCK})(\text{Baud Rate desired})$$

where SCBR is in the range {1, 2, 3, ..., 8191}

Writing a value of zero to BR disables the baud rate generator.

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 (receiver status bits) or writing (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. The following table lists the available choices.

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 — Wake-up 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:															
0	0	0	0	0	0	0	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 register TDR.

TDRE is set when the byte in register TDR is transferred to the transmit serial shifter. If TDRE is zero, transfer has not occurred and a write to TDR will overwrite the previous value. New data is not transmitted if TDR 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 (TDR) of SCDR.

RDRF — Receive Data Register Full Flag

0 = Register RDR is empty or contains previously read data.
1 = Register RDR contains new data.

RDRF is set when the content of the receive serial shifter is transferred to the RDR. 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 RDR, and RDRF is still set. Data transfer is inhibited until OR is cleared. Previous data in RDR 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

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

0 = No framing error 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

1 = Parity error occurred on the received data

0 = No parity error 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. RDR is a read-only register that contains data received by the SCI serial interface. The data comes into the receive serial shifter and is transferred to RDR. TDR is a write-only register that contains data to be transmitted. The data is first written to TDR, 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 Standby RAM Module

This module contains a one Kbyte array of fast (two bus cycle) static RAM, which is especially useful for system stacks and variable storage. SRAM can be mapped to any one Kbyte boundary in the address map, but must not overlap the module control registers (overlap makes the registers inaccessible). Data can be read/written in bytes, words or long words. SRAM is powered by V_{DD} in normal operation. During power-down, SRAM contents are maintained by power from the V_{STBY} input. Power switching between sources is automatic. An address map of the SRAM control registers follows.

Table 19 SRAM Address Map

Address	15 8	7 0
$YFFB00	RAM MODULE CONFIGURATION REGISTER (RAMMCR)	
$YFFB02	RAM TEST REGISTER (RAMTST)	
$YFFB04	RAM ARRAY BASE ADDRESS REGISTER HIGH (RAMBAH)	
$YFFB06	RAM ARRAY BASE ADDRESS REGISTER LOW (RAMBAL)	
$YFFB08	RESERVED	

Y = M111, where M is the logic state of the modmap (MM) bit in the SIMCR

6.1 SRAM Register Block

There are four SRAM control registers: the RAM module configuration register (RAMMCR), the RAM test register (RAMTST), and the RAM array base address registers (RAMBAH/RAMBAL).

There is an 8-byte minimum register block size for the module. Unimplemented register addresses are read as zeros. Writes have no effect.

6.2 SRAM Registers

The CPU16 in the MC68HC16Z1 operates only in supervisory mode. Access to the SRAM array is controlled by the RASP field in RAMMCR. SRAM responds to both program and data space accesses based on the value in the RASP field in RAMMCR. This allows code to be executed from RAM, and permits the use of program counter relative addressing mode for operand fetches from the array.

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

15				11		9	8	7	6	5	4	3	2	1	0
STOP	0	0	0	RLCK	0	RASP		NOT USED							
RESET:															
1	0	0	0	0	0	1	1								

Use RAMMCR to determine whether the RAM is in STOP mode or normal mode. It can also determine in which space the array resides, and controls access to the base array registers. Reads of unimplemented bits always return zeros. Writes do not affect unimplemented bits.

STOP — Stop Control

0 = RAM array operates normally.

1 = RAM array enters low-power stop mode.

This bit controls whether the RAM array is in stop mode or normal operation. Reset state is one, leaving the array configured for LPSTOP operation. In stop mode, the array retains its contents, but cannot be read or written by the CPU. Because the CPU16 operates in supervisor mode, this bit can be read or written at any time.

RLCK — RAM Base Address Lock

0 = SRAM base address registers are writable from IMB

1 = SRAM base address registers are locked

RLCK defaults to zero on reset. It can be written to one once.

RASP[1:0] — RAM Array Space Field

This field limits access to the SRAM array in microcontrollers that support separate user and supervisor operating modes. Because the CPU16 operates in supervisor mode only, RASP1 has no effect.

RASP	Space
X0	Program and Data
X1	Program

RAMTST — RAM Test Register **$YFFB02**

RAMTST is for factory test only. Reads of this register return zeros and writes have no effect.

RAMBAH — Array Base Address Register High **$YFFB04**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NOT USED								ADDR 23*	ADDR 22*	ADDR 21*	ADDR 20*	ADDR 19	ADDR 18	ADDR 17	ADDR 16
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

*ADDR[23:20] is at the same logic level as ADDR19 during internal CPU master operation. ADDR[23:20] must match ADDR19 for the chip select to be active.

RAMBAL — Array Base Address Register Low **$YFFB06**

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

RAMBAH and RAMBAL specify an SRAM base address in the system memory map. They can only be written while the SRAM is in low-power mode (RAMMCR STOP = 1, the default out of reset) and the base address lock is disabled (RAMMCR RLCK = 0, the default out of reset). This prevents accidental remapping of the array. Because the CPU16 drives ADDR[23:20] with the value of ADDR19, the value in the ADDR[23:20] fields must match the value in the ADDR19 field for the array to be accessible.

6.3 SRAM Operation

There are five operating modes.

The RAM module is in normal mode when powered by V_{DD}. The array can be accessed by byte, word, or long word. A byte or aligned word (high-order byte is at an even address) access only takes one bus cycle or two system clocks. A long word or misaligned word access requires two bus cycles.

Standby mode is intended to preserve RAM contents when V_{DD} is removed. SRAM contents are maintained by a power source connected to the V_{STBY} pin. The standby voltage is referred to as V_{SB}. Circuitry within the SRAM module switches to the higher of V_{DD} or V_{SB} with no loss of data. When SRAM is powered from the V_{STBY} pin, access to the array is not guaranteed. If standby operation is not desired, connect the V_{STBY} pin to V_{SS}.

Reset mode allows the CPU to complete the current bus cycle before resetting. When a synchronous reset occurs while a byte or word SRAM access is in progress, the access will be completed. If reset occurs during the first word access of a long-word operation, only the first word access will be completed. If reset occurs during the second word access of a long word operation, the entire access will be completed. Data being read from or written to the RAM may be corrupted by asynchronous reset.

Test mode is used for factory testing of the RAM array.

Writing the STOP bit of RAMMCR causes the SRAM module to enter stop mode. The RAM array is disabled which, if necessary, allows external logic to decode SRAM addresses but all data is retained. If V_{DD} falls below V_{SB}, internal circuitry switches to V_{SB}, as in standby mode. Exit the stop mode by clearing the STOP bit.

7 General-Purpose Timer Module

The GPT is a simple, yet flexible 11-channel timer used in systems where a moderate degree of external visibility and control is required. The GPT consists of two nearly independent submodules, the compare/capture unit, and the pulse-width modulator. Refer to the following block diagram of the GPT.

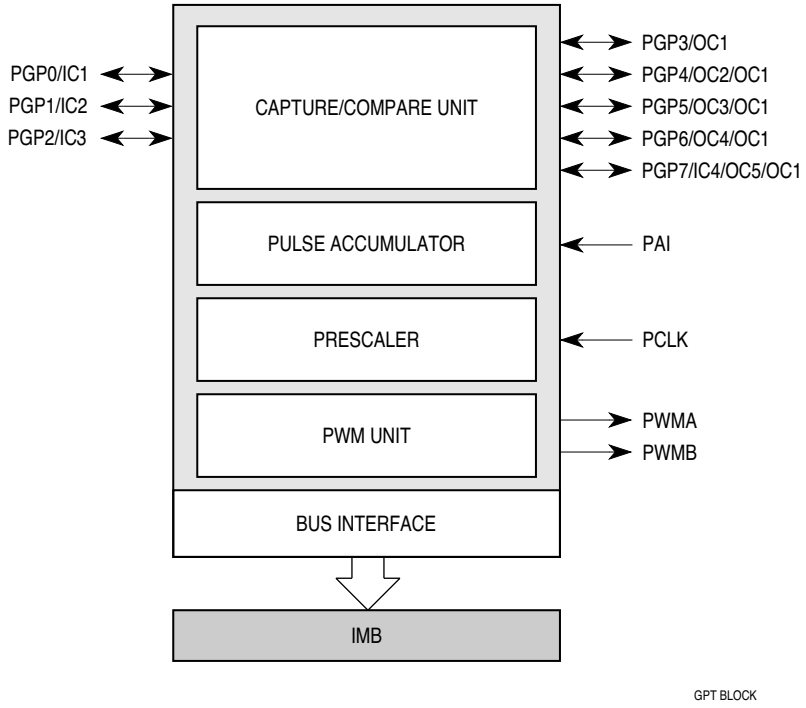


Figure 17 GPT Block Diagram

GPT input capture/output compare pins are bidirectional and can be used to form an 8-bit parallel port. The pulse-width modulator outputs can be used as general-purpose outputs. The PAI and PCLK inputs can be used as general-purpose inputs.

Table 20 GPT Address Map

Address	15 8	7 0
$YFF900	GPT MODULE CONFIGURATION (GPTMCR)	
$YFF902	(RESERVED FOR TEST)	
$YFF904	INTERRUPT CONFIGURATION (ICR)	
$YFF906	PGP DATA DIRECTION (DDRGP)	PGP DATA (PORTGP)
$YFF908	OC1 ACTION MASK (OC1M)	OC1 ACTION DATA (OC1D)
$YFF90A	TIMER COUNTER (TCNT)	
$YFF90C	PA CONTROL (PACTL)	PA COUNTER (PACNT)
$YFF90E	INPUT CAPTURE 1 (TIC1)	
$YFF910	INPUT CAPTURE 2 (TIC2)	
$YFF912	INPUT CAPTURE 3 (TIC3)	
$YFF914	OUTPUT COMPARE 1 (TOC1)	
$YFF916	OUTPUT COMPARE 2 (TOC2)	
$YFF918	OUTPUT COMPARE 3 (TOC3)	
$YFF91A	OUTPUT COMPARE 4 (TOC4)	
$YFF91C	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)	PWM CONTROL C (PWMC)
$YFF926	PWM CONTROL A (PWMA)	PWM CONTROL B (PWMB)
$YFF928	PWM COUNT (PWMCNT)	
$YFF92A	PWMA BUFFER (PWMBUFA)	PWMB BUFFER (PWMBUFB)
$YFF92C	GPT PRESCALER (PRESCL)	
$YFF92E–$YFF93F	RESERVED	

Y = M111, where M is the logic state of the modmap (MM) bit in the SIMCR

7.1 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). These channels share a 16-bit free-running counter (TCNT), which derives its clock from seven stages of a 9-stage prescaler or from external clock input PCLK. 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. The following block diagrams show GPT compare/capture functions and the prescaler.

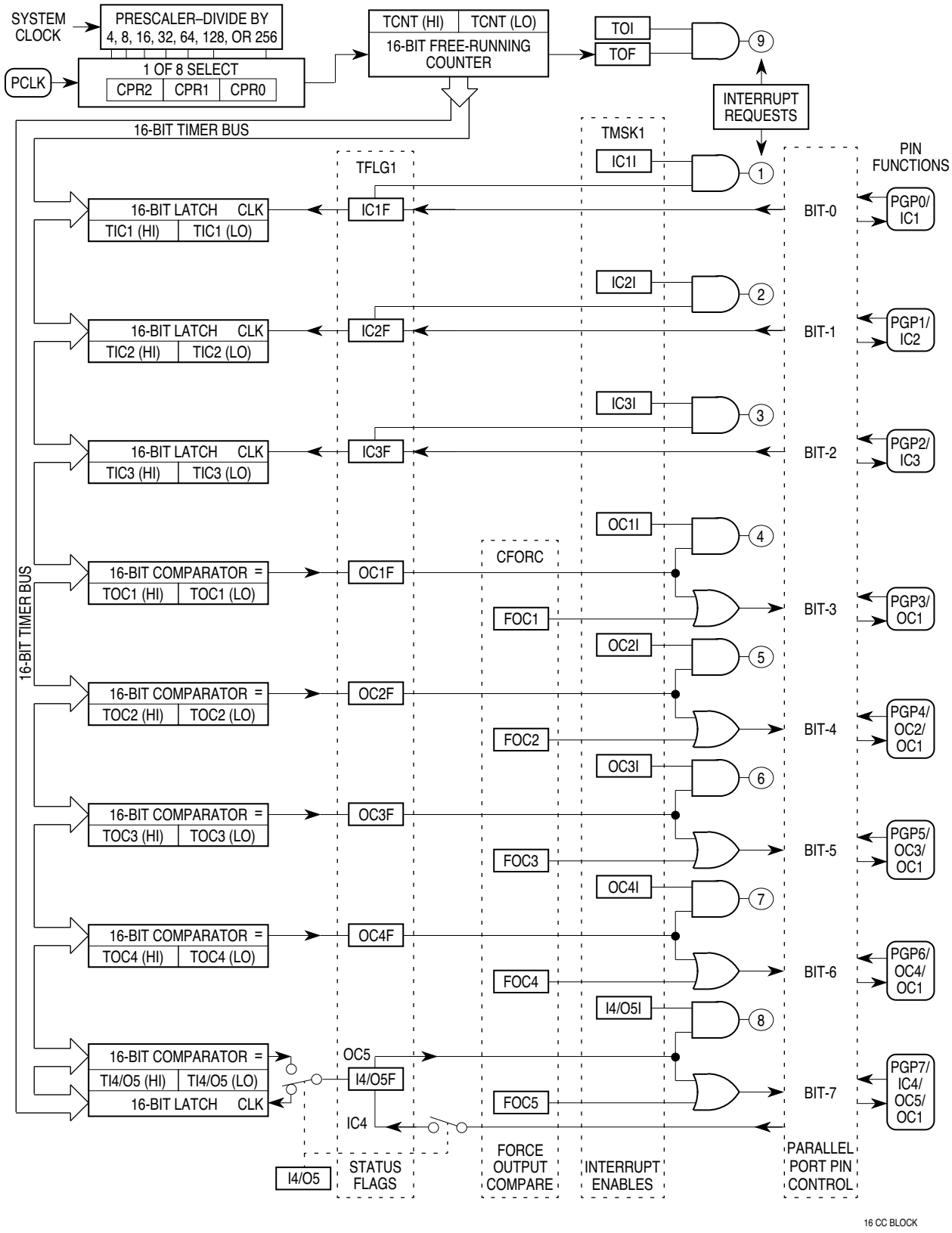


Figure 18 GPT Compare/Capture Block Diagram

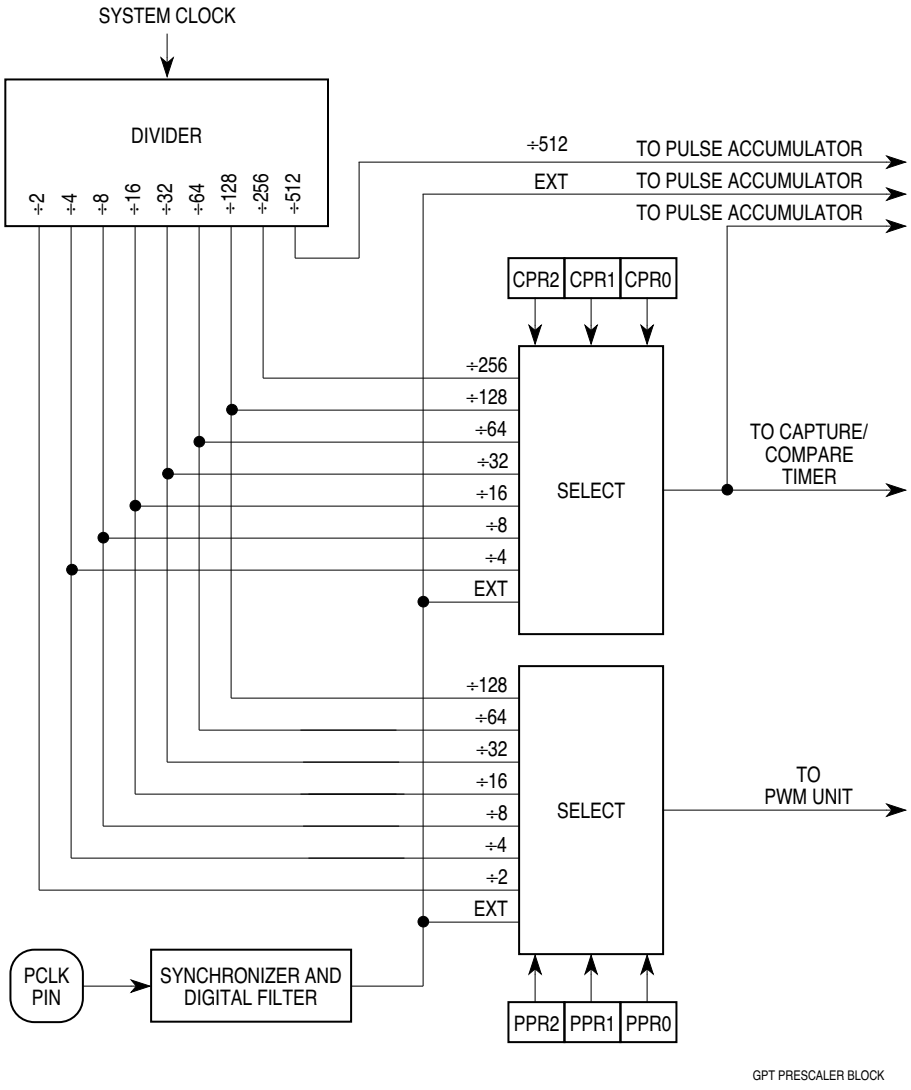


Figure 19 Prescaler Block Diagram

7.2 Pulse-Width Modulator

The pulse-width modulation submodule has two output pins. The outputs are periodic waveforms controlled by a single frequency whose duty cycles can be independently selected and modified by user software. Each PWM can be independently programmed to run in fast or slow mode. The PWM unit has its own 16-bit free-running counter, which is clocked by an output of the nine-stage prescaler (the same prescaler used by the compare/capture unit) or by the clock input pin, PCLK.

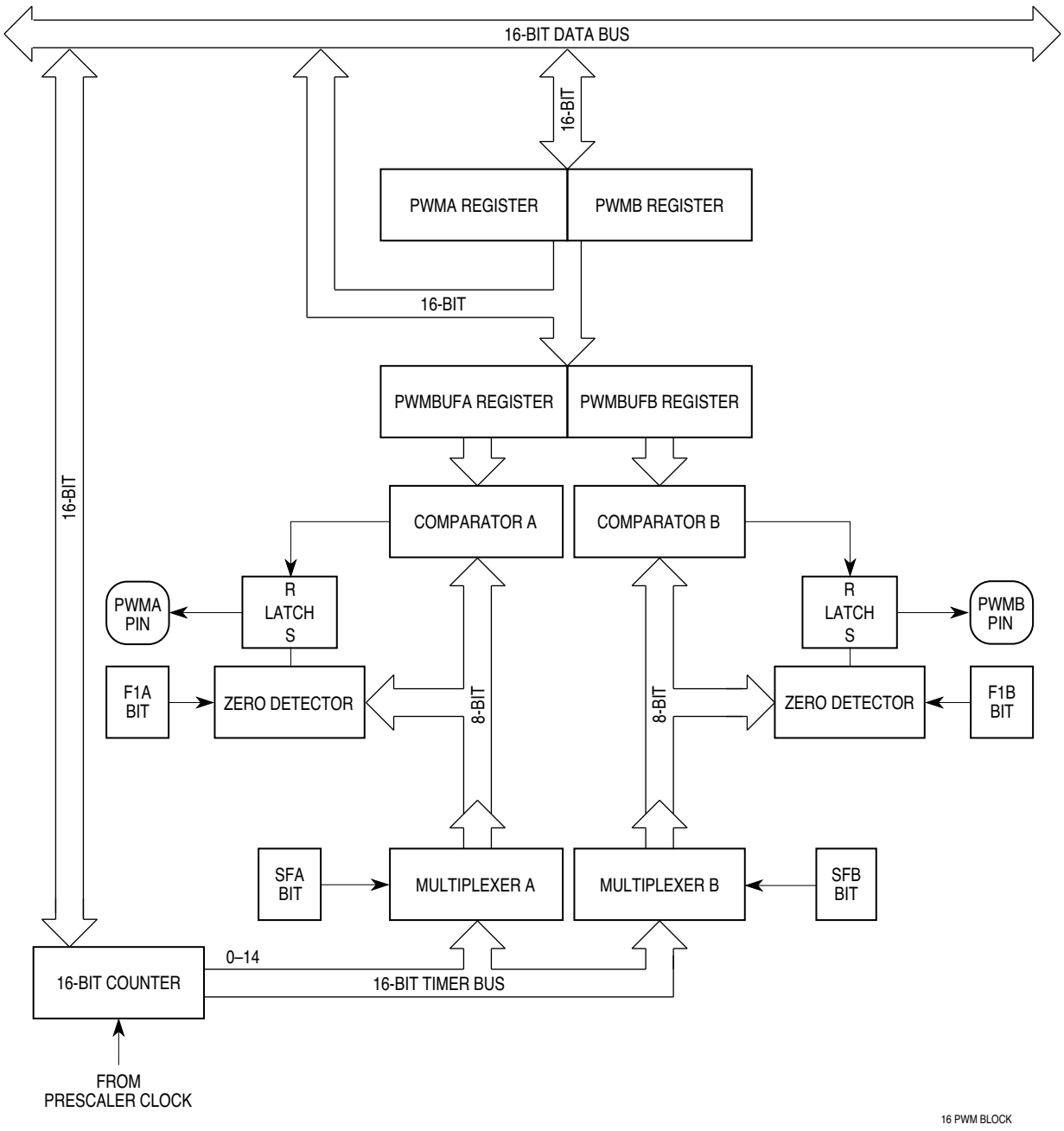


Figure 20 PWM Unit Block Diagram

7.3 GPT Registers

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	IARB3	IARB2	IARB1	IARB0
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 at this time

FRZ0 — FREEZE Response
0 = Ignore FREEZE
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
0 = Registers with access controlled by SUPV are unrestricted (FC2 is a don't care).
1 = Registers with access controlled by SUPV are restricted when FC2 = 1.
Because the CPU16 in the MC68HC16Z1 operates in supervisor mode only (FC2 is always logic level one), this bit has no effect.

IARB[3:0] — Interrupt Arbitration Identification
To enable interrupt arbitration, system software must set this field to a value between $F–$1; $F is the highest priority. This field is initialized to $0 during reset. If the CPU recognizes a GPT interrupt request while IARB = $0, a spurious interrupt exception is taken.

MTR — GPT Module Test Register (Reserved) **$YFF902**

This address is currently unused and returns zeros if read. It 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				0	IPL			IVBA				0	0	0	0
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IPA — Interrupt Priority Adjust
Specifies which GPT interrupt source is given highest internal priority

IPL — Interrupt Priority Level
Specifies the priority level of interrupts generated by the GPT.

IVBA — Interrupt Vector Base Address
Most significant nibble of interrupt vector numbers generated by the GPT.

DDRGP/PORTGP — Port GP Data Direction Register/Port GP Data Register **$YFF906**

15							8	7							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
When PORTGP is used for general-purpose I/O, each bit in the DDRGP determines whether the corresponding PORTGP bit is input or 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
0C1M					0	0	0	0C1D					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
0 = Corresponding output compare pin is not affected by OC1 compare.
1 = Corresponding output compare pin is affected by OC1 compare.
OC1M[5:1] correspond to OC[5:1].

OC1D[5:1] — OC1 Data Field
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.
OC1D[5:1] correspond to OC[5:1].

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		PULSE ACCUMULATOR COUNTER							
RESET:															
U	0	0	0	U	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 System 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 PEDGE and PAMOD are shown in the following table.

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

PCLKS — PCLK Pin State (Read Only)

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)

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.

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	EDGE4		EDGE3		EDGE2		EDGE1	
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.

OM/OL[5:2]	Action Taken
00	Timer Disconnected from Output Logic
01	Toggle OCx Output Line
10	Clear OCx Output Line to 0
11	Set OCx Output Line to 1

EDGE[4:1] — Input Capture Edge Control Bits
Each pair of bits configures input sensing logic for the corresponding input capture.

EDGE[4:1]	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		0
I4/O5I	OCI				ICI			TOI	0	PAOVI	PAII	CPROUT	CPR		
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 flag in TFLG1 is set

OCI[4:1] — Output Compare Interrupt Enable
0 = OC interrupt disabled
1 = OC interrupt requested when OC flag set
OCI[4:1] correspond to OC[4:1].

ICI[3:1] — Input Capture Interrupt Enable
0 = IC interrupt disabled
1 = IC interrupt requested when IC flag set
ICI[3:1] correspond to IC[3:1].

TOI — Timer Overflow Interrupt Enable
0 = Timer overflow interrupt disabled
1 = Interrupt requested when TOF flag is set

PAOVI — Pulse Accumulator Overflow Interrupt Enable
0 = Pulse accumulator overflow interrupt disabled
1 = Interrupt requested when PAOVF flag is set

PAII — Pulse Accumulator Input Interrupt Enable
0 = Pulse accumulator interrupt disabled
1 = Interrupt requested when PAIF flag 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.

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				ICF			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 0, this flag is set each time TCNT matches the value in TOC5. When I4/O5 in PACTL is 1, 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/PWMC — Compare Force Register/PWM Control Register **$YFF924**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FOC					0	FPWMA	FPWMB	PPROUT	PPR			SFA	SFB	F1A	F1B
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 or PWM pins. PWMC sets PWM operating conditions.

FOC[5:1] — Force Output Compare

0 = Has no meaning

1 = Causes pin action programmed for corresponding OC pin, but the OC flag is not set.

FOC[5:1] correspond to OC[5:1].

FPWMA — Force PWMA Value

0 = Normal PWMA operation

1 = The value of F1A is driven out on the PWMA pin, regardless of the state of PPROUT.

FPWMB — Force PWMB Value

0 = Normal PWMB operation

1 = The value of F1B is driven out on the PWMB pin.

PPROUT — PWM Clock Output Enable

0 = Normal PWM operation on PMWA

1 = TCNT clock driven out PWMA pin

PPR[2:0] — PWM Prescaler/PCLK Select

This field selects one of seven prescaler taps or PCLK to be PWMCNT input.

PPR[2:0]	System Clock Divide-by Factor
000	2
001	4
010	8
011	16
100	32
101	64
110	128
111	PCLK

SFA — PWMA Slow/Fast Select

0 = PWMA period is 256 PWMCNT increments long.
1 = PWMA period is 32768 PWMCNT increments long.

SFB — PWMB Slow/Fast Select

0 = PWMB period is 256 PWMCNT increments long.
1 = PWMB period is 32768 PWMCNT increments long.

The following table shows the effects of SF settings on PWM frequency (16.78-MHz system clock).

PPR[2:0]	Prescaler Tap	SFA/B = 0	SFA/B = 1
000	Div 2 = 8.39 MHz	32.8 kHz	256 Hz
001	Div 4 = 4.19 MHz	16.4 kHz	128 Hz
010	Div 8 = 2.10 MHz	8.19 kHz	64.0 Hz
011	Div 16 = 1.05 MHz	4.09 kHz	32.0 Hz
100	Div 32 = 524 kHz	2.05 kHz	16.0 Hz
101	Div 64 = 262 kHz	1.02 kHz	8.0 Hz
110	Div 128 = 131 kHz	512 Hz	4.0 Hz
111	PCLK	PCLK/256	PCLK/32768

F1A — Force Logic Level One on PWMA

0 = Force logic level zero output on PWMA pin
1 = Force logic level one output on PWMA pin

F1B — Force Logic Level One on PWMB

0 = Force logic level zero output on PWMB pin
1 = Force logic level one output on PWMB pin

PWMA/PWMB — PWM Registers A/B **$YFF926, $YFF927**

These registers are associated with the pulse-width value of the PWM output on the corresponding PWM pin. A value of $00 loaded into one of these registers results in a continuously low output on the corresponding pin. A value of $80 results in a 50% duty cycle output. Maximum value ($FF) selects an output that is high for 255/256 of the period.

PWMCNT — PWM Count Register **$YFF928**

PWMCNT is the 16-bit free-running counter associated with the PWM functions of the GPT module.

PWMBUFA/B — PWM Buffer Registers A/B **$YFF92A, $YFF92B**

These read-only registers contain values associated with the duty cycles of the corresponding PWM. Reset state is $0000.

PRESCL — GPT Prescaler **$YFF92C**

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.

8 Electrical Characteristics

This section contains electrical specification tables and reference timing diagrams.

Table 21 Maximum Ratings

Rating	Symbol	Value	Unit
Supply Voltage[1,2,5]	V_{DD}	–0.3 to + 6.5	V
Input Voltage[1,2,3,4,5]	V_{in}	–0.3 to +6.5	V
Instantaneous Maximum Current Single pin limit (applies to all pins)[1,4,5,6]	I_D	25	mA
Operating Maximum Current Digital input disruptive current [4,5,6,7] $V_{SS} - 0.3 \leq V_{IN} \leq V_{DD} + 0.3$	I_{iD}	–500 to 500	µA
Operating Temperature Range MC68HC16Z1 “C” Suffix MC68HC16Z1 “V” Suffix MC68HC16Z1 “M” Suffix	T_A	TL to TH –40 to 85 –40 to 105 –40 to 125	°C
Storage Temperature Range	T_{stg}	–55 to 150	°C

1. Permanent damage can occur if maximum ratings are exceeded. Exposure to voltages or currents in excess of recommended values affects device reliability. Device modules may not operate normally while being exposed to electrical extremes.
2. Although sections of the device contain circuitry to protect against damage from high static voltages or electrical fields, take normal precautions to avoid exposure to voltages higher than maximum-rated voltages
3. Il pins except $\overline{\text{TSTME}}$/TSC.
4. All functional non-supply pins are internally clamped to V_{SS}. All functional pins except EXTAL, $\overline{\text{TSTME}}$/TSC, and XFC are internally clamped to V_{DD}.
5. This parameter is periodically sampled rather than 100% tested.
6. Power supply must maintain regulation within operating V_{DD} range during instantaneous and operating maximum current condition.
7. Total input current for all digital input-only and all digital input/output pins must not exceed 10 mA. Exceeding this limit can cause disruption of normal operation.

Table 22 Thermal Characteristics

Characteristic	Symbol	Value	Unit
Thermal Resistance Plastic 132-Pin Surface Mount Plastic 144-Pin Surface Mount	Θ_{JA}	38	°C/W

The average chip-junction temperature (T_J) in C can be obtained from:

$$T_J = T_A + (P_D \, \Theta_{JA}) \quad (1)$$

where

T_A = Ambient Temperature, °C
Θ_{JA} = Package Thermal Resistance, Junction-to-Ambient, °C/W
P_D = $P_{INT}+P_{I/O}$
P_{INT} = $I_{DD} \times \varsigma_{DD}$, Watts — Chip Internal Power
$P_{I/O}$ = Power Dissipation on Input and Output Pins — User Determined

For most applications $P_{I/O} < P_{INT}$ and can be neglected. An approximate relationship between P_D and T_J (if $P_{I/O}$ is neglected) is:

$$P_D = K \div (T_J + 273°C) \quad (2)$$

Solving equations 1 and 2 for K gives:

$$K = P_D + (T_A + 273°C) + \Theta_{JA} \times \Pi_D{}^2 \quad (3)$$

where K is a constant pertaining to the particular part. K can be determined from equation (3) by measuring P_D (at equilibrium) for a known T_A. Using this value of K, the values of P_D and T_J can be obtained by solving equations (1) and (2) iteratively for any value of T_A.

Table 23 Clock Control Timing

(V_{DD} and V_{DDSYN} = 5.0 Vdc ±10%, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H, 32.768 kHz reference)

Characteristic	Symbol	Min	Max	Unit
PLL Reference Frequency Range	f_{ref}	25	50	kHz
System Frequency[1]	f_{sys}	dc	16.78	MHz
On-Chip PLL Frequency		0.131	16.78	
External Clock Operation		dc	16	
PLL Lock Time[2]	t_{lpll}	—	20	ms
Limp Mode Clock Frequency[3]	f_{limp}			MHz
SYNCR X bit = 0		—	f_{sys} max/2	
SYNCR X bit = 1		—	f_{sys} max	
CLKOUT Stability[4,5]	C_{stab}			%
Short term		–1.0	1.0	
Long term		–0.5	0.5	

1. All internal registers retain data at 0 Hz.
2. Assumes that stable V_{DDSYN} is applied, that an external filter capacitor with a value of 0.1 µF is attached to the XFC pin, and that the crystal oscillator is stable. Lock time is measured from power-up to $\overline{\text{RESET}}$ release. This specification also applies to the period required for PLL lock after changing the W and Y frequency control bits in the synthesizer control register (SYNCR) while the PLL is running, and to the period required for the clock to lock after LPSTOP.
3. Determined by the initial control voltage applied to the on-chip VCO. The X bit in SYNCR controls a divide by two scaler on the system clock output.
4. Short-term CLKOUT stability is the average deviation from programmed frequency measured over a 2 µs interval at maximum f_{sys}. Long-term CLKOUT stability is the average deviation from programmed frequency measured over a 1 ms interval at maximum f_{sys}. Stability is measured with a stable external clock applied — variation in crystal oscillator frequency is additive to this figure.
5. This parameter is periodically sampled rather than 100% tested.

Table 24 DC Characteristics

(V_{DD} and V_{DDSYN} = 5.0 Vdc ± 10%, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H)

Characteristic	Symbol	Min	Max	Unit
Input High Voltage	V_{IH}	0.7 (ς_{DD})	V_{DD} + 0.3	V
Input Low Voltage	V_{IL}	V_{SS} – 0.3	0.2 (ς_{DD})	V
Input Hysteresis[1,9]	V_{HYS}	0.5	—	V
Input Leakage Current[2] V_{in} =V_{DD} or V_{SS} All input-only pins except ADC pins	I_{in}	–2.5	2.5	µA
High Impedance (Off-State) Leakage Current[2] V_{in} =V_{DD} or V_{SS} All input/output and output pins	I_{OZ}	–2.5	2.5	µA
CMOS Output High Voltage[2,3] I_{OH} = –10.0 µA Group 1, 2, 4 input/output and all output pins	V_{OH}	V_{DD} – 0.2	—	V
CMOS Output Low Voltage[2] I_{OL} = 10.0 µA Group 1, 2, 4 input/output and all output pins	V_{OL}	—	0.2	V
Output High Voltage [2,3] I_{OH} =–0.8 mA Group 1, 2, 4 input/output and all output pins	V_{OH}	V_{DD} – 0.8	—	V
Output Low Voltage[2] I_{OL} = 1.6 mA Group 1 I/O Pins, CLKOUT, FREEZE/QUOT, IPIPE0 I_{OL} = 5.3 mA Group 2 and Group 4 I/O Pins, $\overline{CSBOOT}$, $\overline{BG}/\overline{CS}$ I_{OL} = 12 mA Group 3	V_{OL}	— — —	0.4 0.4 0.4	V
Three State Control Input High Voltage	V_{IHTSC}	1.6 (V_{DD})	9.1	V
Data Bus Mode Select Pull-up Current[5] V_{in} = V_{IL} DATA[15:0] V_{in} = V_{IH} DATA[15:0]	I_{MSP}	— –15	–120 —	µA
V_{DD} Supply Current[6] RUN[4] LPSTOP, 32.768 kHz crystal, VCO Off (STSIM = 0) LPSTOP (External clock input frequency = maximum f_{sys})	I_{DD} S_{IDD} S_{IDD}	— — —	110 350 5	mA µA mA
Clock Synthesizer Operating Voltage	V_{DDSYN}	4.5	5.5	V
V_{DDSYN} Supply Current[6] 32.768 kHz crystal, VCO on, maximum f_{sys} External Clock, maximum f_{sys} LPSTOP, 32.768 kHz crystal, VCO off (STSIM = 0) 32.768 kHz crystal, V_{DD} powered down	I_{DDSYN} I_{DDSYN} S_{IDDSYN} I_{DDSYN}	— — — —	1 5 150 100	mA mA µA µA
RAM Standby Voltage[7] Specified V_{DD} applied V_{DD} = V_{SS}	V_{SB}	0.0 3.0	5.5 5.5	V
RAM Standby Current[7] Specified V_{DD} applied V_{DD} = V_{SS}	I_{SB} I_{SB}	–2.5 —	2.5 50	µA µA
Power Dissipation[8]	P_D	—	605	mW
Input Capacitance[2,9] All input-only pins except ADC pins All input/output pins	C_{in}	— —	10 20	pF
Load Capacitance[2] Group 1 I/O Pins and CLKOUT, FREEZE/QUOT, IPIPE0 Group 2 I/O Pins and $\overline{CSBOOT}$, $\overline{BG}/\overline{CS}$ Group 3 I/O pins Group 4 I/O pins	C_L	— — — —	90 100 130 200	pF

NOTES:

1. Applies to:
   - Port ADA [7:0] — AN[7:0]
   - Port E [7:4] SIZ[1:0], $\overline{AS}$, $\overline{DS}$
   - Port F [7:0] $\overline{IRQ}$[7:1], MODCLK
   - Port GP [7:0] — IC4/OC5/OC1, IC[3:1], OC[4:1]/OC1
   - Port QS [7:0] — TXD, PCS[3:1],ÊPCS0/$\overline{SS}$, SCK, MOSI, MISO
   - $\overline{BKPT}$/DSCLK, DSI/IPIPE1, PAI, PCLK, $\overline{RESET}$, RXD, $\overline{TSTME}$/TSC

2. Input-Only Pins: $\overline{TSTME}$/TSC, $\overline{BKPT}$/DSCLK, PAI, PCLK, RXD
   Output-Only Pins: $\overline{CSBOOT}$, $\overline{BG}$/$\overline{CS1}$, CLKOUT, FREEZE/QUOT, DS0/IPIPE0, PWMA, PWMB
   Input/Output Pins:

Group 1:	Port GP [7:0] — IC4/OC5/OC1, IC[3:1], OC[4:1]/OC1
	DATA[15:0], DSI/IPIPE1
Group 2:	Port C [6:0] — ADDR[22:19]/$\overline{CS[9:6]}$, FC[2:0]/$\overline{CS[5:3]}$
	Port E [7:0] —SIZ[1:0], $\overline{AS}$, $\overline{DS}$, $\overline{AVEC}$, $\overline{DSACK[1:0]}$
	Port F [7:0] — $\overline{IRQ}$[7:1], MODCLK
	Port QS [7:3] — TXD, PCS[3:1],ÊPCS0/$\overline{SS}$
	ADDR23/$\overline{CS10}$/ECLK, ADDR[18:0], R/$\overline{W}$, $\overline{BERR}$, $\overline{BR}$/$\overline{CS0}$, $\overline{BGACK}$/$\overline{CS2}$
Group 3:	$\overline{HALT}$, $\overline{RESET}$
Group 4:	MISO, MOSI, SCK

3. Does not apply to $\overline{HALT}$ and $\overline{RESET}$ because they are open drain pins. Does not apply to Port QS [7:0] (TXD, PCS[3:1],ÊPCS0/$\overline{SS}$, SCK, MOSI, MISO) in wired-OR mode.
4. Current measured with system clock frequency of 16.78 MHz, all modules active.
5. Use of an active pulldown device is recommended.
6. Total operating current is the sum of the appropriate V_{DD} supply and V_{DDSYN} supply currents.
7. The SRAM module will not switch into standby mode as long as V_{SB} does not exceed V_{DD} by more than 0.5 Volt. The SRAM array cannot be accessed while the module is in standby mode.
8. Power dissipation measured with system clock frequency of 16.78 MHz, all modules active. Power dissipation is calculated using the following expression:

   P_D = Maximum V_{DD} (I_{DDSYN} + I_{DD})

9. This parameter is periodically sampled rather than 100% tested.

Table 25 AC Timing

(V_{DD} and V_{DDSYN} = 5.0 Vdc ± 10%, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H)

Num	Characteristic	Symbol	Min	Max	Unit
F1[2]	Frequency of Operation (32.768 kHz crystal)	f	0.13	16.78	MHz
1	Clock Period	t_{cyc}	59.6	—	ns
1A	ECLK Period	t_{Ecyc}	476	—	ns
1B[3]	External Clock Input Period	t_{Xcyc}	64	—	ns
2, 3	Clock Pulse Width	t_{CW}	24	—	ns
2A, 3A	ECLK Pulse Width	t_{ECW}	236	—	ns
2B, 3B[3]	External Clock Input High/Low Time	t_{XCHL}	32	—	ns
4, 5	CLKOUT Rise and Fall Time	t_{Crf}	—	5	ns
4A, 5A	Rise and Fall Time (All Outputs except CLKOUT)	t_{rf}	—	8	ns
4B, 5B	External Clock Input Rise and Fall Time	t_{XCrf}	—	5	ns
6	Clock High to ADDR, FC, SIZE Valid	t_{CHAV}	0	29	ns
7	Clock High to ADDR, DATA, FC, SIZE High Impedance	t_{CHAZx}	0	59	ns
8	Clock High to ADDR, FC, SIZE, Invalid	t_{CHAZn}	0	—	ns
9	Clock Low to $\overline{AS}$, $\overline{DS}$, $\overline{CS}$ Asserted	t_{CLSA}	2	25	ns
9A[4]	$\overline{AS}$ to $\overline{DS}$ or $\overline{CS}$ Asserted (Read)	t_{STSA}	–15	15	ns
11	ADDR, FC, SIZE Valid to $\overline{AS}$, $\overline{CS}$, (and $\overline{DS}$ Read) Asserted	t_{AVSA}	15	—	ns
12	Clock Low to $\overline{AS}$, $\overline{DS}$, $\overline{CS}$ Negated	t_{CLSN}	2	29	ns
13	$\overline{AS}$, $\overline{DS}$, $\overline{CS}$ Negated to ADDR, FC, SIZE Invalid (Address Hold)	t_{SNAI}	15	—	ns
14	$\overline{AS}$, $\overline{CS}$ Width Asserted	t_{SWA}	100	—	ns
14A	$\overline{DS}$, $\overline{CS}$ Width Asserted (Write)	t_{SWAW}	45	—	ns
14B	$\overline{AS}$, $\overline{CS}$ Width Asserted (Fast Cycle)	t_{SWDW}	40	—	ns
15[5]	$\overline{AS}$, $\overline{DS}$, $\overline{CS}$ Width Negated	t_{SN}	40	—	ns
16	Clock High to $\overline{AS}$, $\overline{DS}$, R/$\overline{W}$ High Impedance	t_{CHSZ}	—	59	ns
17	$\overline{AS}$, $\overline{DS}$, $\overline{CS}$ Negated to R/$\overline{W}$ High	t_{SNRN}	15	—	ns
18	Clock High to R/$\overline{W}$ High	t_{CHRH}	0	29	ns
20	Clock High to R/$\overline{W}$ Low	t_{CHRL}	0	29	ns
21	R/$\overline{W}$ High to $\overline{AS}$, $\overline{CS}$ Asserted	t_{RAAA}	15	—	ns
22	R/$\overline{W}$ Low to $\overline{DS}$, $\overline{CS}$ Asserted (Write)	t_{RASA}	70	—	ns
23	Clock High to Data Out Valid	t_{CHDO}	—	29	ns
24	Data Out Valid to Negating Edge of $\overline{AS}$, $\overline{CS}$	t_{DVASN}	15	—	ns
25	$\overline{DS}$, $\overline{CS}$ Negated to Data Out Invalid (Data Out Hold)	t_{SNDOI}	15	—	ns
26	Data Out Valid to $\overline{DS}$, $\overline{CS}$ Asserted (Write)	t_{DVSA}	15	—	ns
27	Data In Valid to Clock Low (Data Setup)	t_{DICL}	5	—	ns
27A	Late $\overline{BERR}$, $\overline{HALT}$ Asserted to Clock Low (Setup Time)	t_{BELCL}	20	—	ns
28	$\overline{AS}$, $\overline{DS}$ Negated to $\overline{DSACKx}$, $\overline{BERR}$, $\overline{HALT}$, $\overline{AVEC}$ Negated	t_{SNDN}	0	80	ns
29[6]	$\overline{DS}$, $\overline{CS}$ Negated to Data In Invalid (Data In Hold)	t_{SNDI}	0	—	ns
29A[6, 7]	$\overline{DS}$, $\overline{CS}$ Negated to Data In High Impedance	t_{SHDI}	—	55	ns
30[6]	CLKOUT Low to Data In Invalid (Fast Cycle Hold)	t_{CLDI}	15	—	ns
30A[6]	CLKOUT Low to Data In High Impedance	t_{CLDH}	—	90	ns
31[8]	$\overline{DSACKx}$ Asserted to Data In Valid	t_{DADI}	—	50	ns
33	Clock Low to $\overline{BG}$ Asserted/Negated	t_{CLBAN}	—	29	ns

Table 25 AC Timing (Continued)

(V_{DD} and V_{DDSYN} = 5.0 Vdc ± 10%, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H)

Num	Characteristic	Symbol	Min	Max	Unit
35[9]	$\overline{BR}$ Asserted to $\overline{BG}$ Asserted	t_{BRAGA}	1	—	t_{cyc}
37	$\overline{BGACK}$ Asserted to $\overline{BG}$ Negated	t_{GAGN}	1	2	t_{cyc}
39	$\overline{BG}$ Width Negated	t_{GH}	2	—	t_{cyc}
39A	$\overline{BG}$ Width Asserted	t_{GA}	1	—	t_{cyc}
46	R/$\overline{W}$ Width Asserted (Write or Read)	t_{RWA}	150	—	ns
46A	R/$\overline{W}$ Width Asserted (Fast Write or Read Cycle)	t_{RWAS}	90	—	ns
47A	Asynchronous Input Setup Time $\overline{BR}$, $\overline{BGACK}$, $\overline{DSACKx}$, $\overline{BERR}$, $\overline{AVEC}$, $\overline{HALT}$	t_{AIST}	5	—	ns
47B	Asynchronous Input Hold Time	t_{AIHT}	15	—	ns
48[10]	$\overline{DSACKx}$ Asserted to $\overline{BERR}$, $\overline{HALT}$ Asserted	t_{DABA}	—	30	ns
53	Data Out Hold from Clock High	t_{DOCH}	0	—	ns
54	Clock High to Data Out High Impedance	t_{CHDH}	—	28	ns
55	R/$\overline{W}$ Asserted to Data Bus Impedance Change	t_{RADC}	40	—	ns
70	Clock Low to Data Bus Driven (Show Cycle)	t_{SCLDD}	0	29	ns
71	Data Setup Time to Clock Low (Show Cycle)	t_{SCLDS}	15	—	ns
72	Data Hold from Clock Low (Show Cycle)	t_{SCLDH}	10	—	ns
73	$\overline{BKPT}$ Input Setup Time	t_{BKST}	15	—	ns
74	$\overline{BKPT}$ Input Hold Time	t_{BKHT}	10	—	ns
75	Mode Select Setup Time	t_{MSS}	20	—	t_{cyc}
76	Mode Select Hold Time	t_{MSH}	0	—	ns
77	$\overline{RESET}$ Assertion Time[11]	t_{RSTA}	4	—	t_{cyc}
78	$\overline{RESET}$ Rise Time[12]	t_{RSTR}	—	10	t_{cyc}
100	CLKOUT High to Phase 1 Asserted[13]	t_{CHP1A}	3	40	ns
101	CLKOUT High to Phase 2 Asserted[13]	t_{CHP2A}	3	40	ns
102	Phase 1 Valid to $\overline{AS}$ or $\overline{DS}$ Asserted[13]	t_{P1VSA}	—	10	ns
103	Phase 2 Valid to $\overline{AS}$ or $\overline{DS}$ Negated[13]	t_{P2VSN}	—	10	ns
104	$\overline{AS}$ or $\overline{DS}$ Valid to Phase 1 Asserted[13]	t_{SAP1A}	—	10	ns
105	$\overline{AS}$ or $\overline{DS}$ Negated to Phase 2 Negated[13]	t_{SNP2N}	—	10	ns

NOTES:

1. All AC timing is shown with respect to 20% V_{DD} and 70% V_{DD} levels unless otherwise noted.
2. Minimum system clock frequency is four times the crystal frequency, subject to specified limits.
3. Minimum external clock high and low times are based on a 50% duty cycle. The minimum allowable t_{Xcyc} period will be reduced when the duty cycle of the external clock signal varies. The relationship between external clock input duty cycle and minimum t_{Xcyc} is expressed:

   Minimum t_{Xcyc} period = minimum t_{XCHL} / (50% – external clock input duty cycle tolerance).

   To achieve maximum operating frequency (f_{sys}) while using an external clock input, adjust clock input duty cycle to obtain a 50% duty cycle on CLKOUT.
4. Specification 9A is the worst-case skew between $\overline{AS}$ and $\overline{DS}$ or $\overline{CS}$. The amount of skew depends on the relative loading of these signals. When loads are kept within specified limits, skew will not cause $\overline{AS}$ and $\overline{DS}$ to fall outside the limits shown in specification 9.
5. If multiple chip selects are used, $\overline{CS}$ width negated (specification 15) applies to the time from the negation of a heavily loaded chip select to the assertion of a lightly loaded chip select. The $\overline{CS}$ width negated specification between multiple chip selects does not apply to chip selects being used for synchronous ECLK cycles.
6. Hold times are specified with respect to $\overline{DS}$ or $\overline{CS}$ on asynchronous reads and with respect to CLKOUT on fast cycle reads. The user is free to use either hold time.
7. Maximum value is equal to (t_{cyc} / 2) + 25 ns.
8. If the asynchronous setup time (specification 47A) requirements are satisfied, the $\overline{DSACKx}$ low to data setup time (specification 31) and $\overline{DSACKx}$ low to $\overline{BERR}$ low setup time (specification 48) can be ignored. The data must only satisfy the data-in to clock low setup time (specification 27) for the following clock cycle. $\overline{BERR}$ must satisfy only the late $\overline{BERR}$ low to clock low setup time (specification 27A) for the following clock cycle.
9. To ensure coherency during every operand transfer, $\overline{BG}$ is not asserted in response to $\overline{BR}$ until after all cycles of the current operand transfer are complete.
10. In the absence of $\overline{DSACKx}$, $\overline{BERR}$ is an asynchronous input using the asynchronous setup time (specification 47A).
11. After external $\overline{RESET}$ negation is detected, a short transition period (approximately 2 t_{cyc}) elapses, then the SIM drives $\overline{RESET}$ low for 512 t_{cyc}.
12. External logic must pull $\overline{RESET}$ high during this period in order for normal MCU operation to begin.
13. Eight pipeline states are multiplexed into IPIPE[1:0]. The multiplexed signals have two phases.
14. Address access time = (2.5 + WS) t_{cyc} – t_{CHAV} – t_{DICL}

    Chip select access time = (2 + WS) t_{cyc} – t_{CLSA} – t_{DICL}

    Where: WS = number of wait states. When fast termination is used (2 clock bus) WS = –1.

Table 26 Background Debugging Mode Timing

(V_{DD} = 5.0 Vdc ± 10%, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H)

Num	Characteristic	Symbol	Min	Max	Unit
B0	DSI Input Setup Time	t_{DSISU}	15	—	ns
B1	DSI Input Hold Time	t_{DSIH}	10	—	ns
B2	DSCLK Setup Time	t_{DSCSU}	15	—	ns
B3	DSCLK Hold Time	t_{DSCH}	10	—	ns
B4	DSO Delay Time	t_{DSOD}	—	25	ns
B5	DSCLK Cycle Time	t_{DSCCYC}	2	—	t_{cyc}
B6	CLKOUT High to FREEZE Asserted/Negated	t_{FRZAN}	—	50	ns
B7	CLKOUT High to IPIPE1 High Impedance	t_{IFZ}	—	50	ns
B8	CLKOUT High to IPIPE1 Valid	t_{IF}	—	50	ns
B9	DSCLK Low Time	t_{DSCLO}	1	—	t_{cyc}

NOTES:
1. All AC timing is shown with respect to 20% V_{DD} and 70% V_{DD} levels unless otherwise noted.

Table 27 ECLK Bus Timing

(V_{DD} = 5.0 Vdc ± 10%, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H)

Num	Characteristic	Symbol	Min	Max	Unit
E1[2]	ECLK Low to Address Valid	t_{EAD}	—	60	ns
E2	ECLK Low to Address Hold	t_{EAH}	10	—	ns
E3	ECLK Low to $\overline{CS}$ Valid ($\overline{CS}$ delay)	t_{ECSD}	—	150	ns
E4	ECLK Low to $\overline{CS}$ Hold	t_{ECSH}	15	—	ns
E5	$\overline{CS}$ Negated Width	t_{ECSN}	30	—	ns
E6	Read Data Setup Time	t_{EDSR}	30	—	ns
E7	Read Data Hold Time	t_{EDHR}	15	—	ns
E8	ECLK Low to Data High Impedance	t_{EDHZ}	—	60	ns
E9	$\overline{CS}$ Negated to Data Hold (Read)	t_{ECDH}	0	—	ns
E10	$\overline{CS}$ Negated to Data High Impedance	t_{ECDZ}	—	1	t_{cyc}
E11	ECLK Low to Data Valid (Write)	t_{EDDW}	—	2	t_{cyc}
E12	ECLK Low to Data Hold (Write)	t_{EDHW}	5	—	ns
E13	$\overline{CS}$ Negated to Data Hold (Write)	t_{ECHW}	0	—	ns
E14[3]	Address Access Time (Read)	t_{EACC}	386	—	ns
E15[4]	Chip Select Access Time (Read)	t_{EACS}	296	—	ns
E16	Address Setup Time	t_{EAS}	—	1/2	t_{cyc}

NOTES:
1. All AC timing is shown with respect to 20% V_{DD} and 70% V_{DD} levels unless otherwise noted.
2. When previous bus cycle is not an ECLK cycle, the address may be valid before ECLK goes low.
3. Address access time = $t_{Ecyc} - t_{EAD} - t_{EDSR}$
4. Chip select access time = $t_{Ecyc} - t_{ECSD} - t_{EDSR}$

Table 28 QSPI Timing

(V_{DD} = 5.0 Vdc ± 10%, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H, 200 pF load on all QSPI pins)

Num	Function	Symbol	Min	Max	Unit
	Operating Frequency Master Slave	f_{op}	 DC DC	 1/4 1/4	 System Clock Frequency System Clock Frequency
1	Cycle Time Master Slave	t_{qcyc}	 4 4	 510 —	 t_{cyc} t_{cyc}
2	Enable Lead Time Master Slave	t_{lead}	 2 2	 128 —	 t_{cyc} t_{cyc}
3	Enable Lag Time Master Slave	t_{lag}	 — 2	 1/2 —	 SCK t_{cyc}
4	Clock (SCK) High or Low Time Master Slave[2]	t_{sw}	 2 t_{cyc} – 60 2 t_{cyc} – n	 255 t_{cyc} —	 ns ns
5	Sequential Transfer Delay Master Slave (Does Not Require Deselect)	t_{td}	 17 13	 8192 —	 t_{cyc} t_{cyc}
6	Data Setup Time (Inputs) Master Slave	t_{su}	 30 20	 — —	 ns ns
7	Data Hold Time (Inputs) Master Slave	t_{hi}	 0 20	 — —	 ns ns
8	Slave Access Time	t_a	—	1	t_{cyc}
9	Slave MISO Disable Time	t_{dis}	—	2	t_{cyc}
10	Data Valid (after SCK Edge) Master Slave	t_v	 — —	 50 50	 ns ns
11	Data Hold Time (Outputs) Master Slave	t_{ho}	 0 0	 — —	 ns ns
12	Rise Time Input Output	 t_{ri} t_{ro}	 — —	 2 30	 μσ νσ
13	Fall Time Input Output	 t_{fi} t_{fo}	 — —	 2 30	 μσ νσ

NOTES:
1. All AC timing is shown with respect to 20% V_{DD} and 70% V_{DD} levels unless otherwise noted.
2. In formula, n = External SCK rise + External SCK fall time.

Table 29 ADC Maximum Ratings

Num	Parameter	Symbol	Min	Max	Unit
1	Analog Supply	V_{DDA}	– 0.3	6.5	V
2	Internal Digital Supply	V_{DDI}	– 0.3	6.5	V
3	Reference Supply	V_{RH}, V_{RL}	– 0.3	6.5	V
4	V_{SS} Differential Voltage	$V_{SSI} - V_{SSA}$	– 0.1	0.1	V
5	V_{DD} Differential Voltage	$V_{DDI} - V_{DDA}$	– 6.5	6.5	V
6	V_{REF} Differential Voltage	$V_{RH} - V_{RL}$	– 6.5	6.5	V
7	V_{REF} to V_{DDA} Differential Voltage	$V_{RH} - V_{DDA}$	– 6.5	6.5	V
8	Disruptive Input Current[1, 2, 3, 4] $V_{SSA} - 0.3 \leq V_{INA} \leq V_{DDA} + 2$	I_{NA}	– 15	15	µA
9	Maximum Input Current[5, 3] $V_{SSA} - 1 \leq V_{INA} \leq V_{DDA} + 3.5$	I_{MA}	– 500	500	µA

1. Below disruptive current conditions, the channel being stressed will have conversion values of $3FF for analog inputs greater than V_{RH} and $000 for values less than V_{RL}. This assumes that $V_{RH} \leq V_{DDA}$ and $V_{RL} \geq V_{SSA}$ due to the presence of the sample amplifier. Other channels are not affected by non-disruptive conditions.
2. Input signals with large slew rates or high frequency noise components cannot be converted accurately. These signals also interfere with conversion of other channels.
3. This parameter is periodically sampled rather than 100% tested.
4. Applies to single pin only.
5. Exceeding limit may cause conversion error on stressed channels and on unstressed channels. Transitions within the limit do not affect device reliability or cause permanent damage.

Table 30 ADC DC Electrical Characteristics (Operating)

(V_{SS} = 0 Vdc, ADCLK = 2.1 MHz, T_A within operating temperature range)

Num	Parameter	Symbol	Min	Max	Unit
1	Analog Supply[1]	V_{DDA}	4.5	5.5	V
2	Internal Digital Supply[1]	V_{DDI}	4.5	5.5	V
3	V_{SS} Differential Voltage	$V_{SSI} - V_{SSA}$	– 1.0	1.0	mV
4	V_{DD} Differential Voltage	$V_{DDI} - V_{DDA}$	– 1.0	1.0	V
5	Reference Voltage Low[2,3]	V_{RL}	V_{SSA}	V_{DDA} / 2	V
6	Reference Voltage High[2,3]	V_{RH}	V_{DDA} / 2	V_{DDA}	V
7	V_{REF} Differential Voltage[3]	$V_{RH} - V_{RL}$	4.5	5.5	V
8	Input Voltage[2]	V_{INDC}	V_{SSA}	V_{DDA}	V
9	Input High, Port ADA	V_{IH}	0.7 (V_{DDA})	V_{DDA} + 0.3	V
10	Input Low, Port ADA	V_{IL}	V_{SSA} – 0.3	0.2 (V_{DDA})	V
15	Analog Supply Current[4]	I_{DDA}	—	1.0	mA
16	Analog Supply Current, LPSTOP	S_{DDA}	—	TBD	µA
17	Reference Supply Current	I_{REF}	—	250	µA
18	Input Current, Off Channel[5]	I_{OFF}	—	250	nA
19	Total Input Capacitance, Not Sampling	C_{INN}	—	10	pF
20	Total Input Capacitance, Sampling	C_{INS}	—	15	pF

1. Refers to operation over full temperature and frequency range.
2. To obtain full-scale, full-range results, $V_{SSA} \le V_{RL} \le V_{INDC} \le V_{RH} \le V_{DDA}$.
3. Accuracy tested and guaranteed at $V_{RH} - V_{RL} \le 5.0\ V \pm 10\%$.
4. Current measured at maximum system clock frequency with ADC active.
5. Maximum leakage occurs at maximum operating temperature. Current decreases by approximately one-half for each 10° C decrease from maximum temperature.

Table 31 ADC AC Characteristics (Operating)

(V_{DD} and V_{DDA} = 5.0 Vdc ± 10%, V_{SS} = 0 Vdc, T_A within operating temperature range)

Num	Parameter	Symbol	Min	Max	Unit
1	IMB Clock Frequency	F_{ICLK}	2.0	16.78	MHz
2	ADC Clock Frequency	F_{ADCLK}	0.5	2.1	MHz
3	8-bit Conversion Time (16 ADC Clocks)[1]	T_{CONV}	7.62	—	µs
4	10-bit Conversion Time (18 ADC Clocks)[1]	T_{CONV}	8.58	—	µs
5	Stop Recovery Time	T_{SR}	—	10	µs

1. Assumes 2.1 MHz ADC clock and selection of minimum sample time (2 ADC clocks).

Table 32 ADC Conversion Characteristics (Operating)

(V_{DD} and V_{DDA} = 5.0 Vdc ± 10%, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H, ADCLK = 2.1 MHz)

Num	Parameter	Symbol	Min	Typ	Max	Unit
1	8-bit Resolution[1]	1 Count	—	20	—	mV
2	8-bit Differential Nonlinearity[2]	DNL	–.5	—	.5	Counts
3	8-bit Integral Nonlinearity[2]	INL	–1	—	1	Counts
4	8-bit Absolute Error[2,3]	AE	–1	—	1	Counts
5	10-bit Resolution[1]	1 Count	—	5	—	mV
6	10-bit Differential Nonlinearity[2]	DNL	–1	—	1	Counts
7	10-bit Integral Nonlinearity[2]	INL	–2	—	2	Counts
8	10-bit Absolute Error[2,4]	AE	–2.5	—	2.5	Counts
9	Source Impedance at Input[5]	R_S	—	20	See Note 5	kΩ

1. $V_{RH} - V_{RL} \geq 5.12$ V; $V_{DDA} - V_{SSA} = 5.12$ V
2. At V_{REF} = 5.12 V, one 10-bit count = 5 mV and one 8-bit count = 20 mV.
3. 8-bit absolute error of 1 count (20 mV) includes 1/2 count (10 mV) inherent quantization error and 1/2 count (10 mV) circuit (differential, integral, and offset) error.
4. 10-bit absolute error of 2.5 counts (12.5 mV) includes 1/2 count (2.5 mV) inherent quantization error and 2 counts (10 mV) circuit (differential, integral, and offset) error.
5. Maximum source impedance is application-dependent. Error resulting from pin leakage depends on junction leakage into the pin and on leakage due to charge-sharing with internal capacitance. In the following expressions, expected error in result value due to leakage is expressed in voltage (V_{errx}).

   Error from junction leakage is a function of external source impedance and input leakage current:

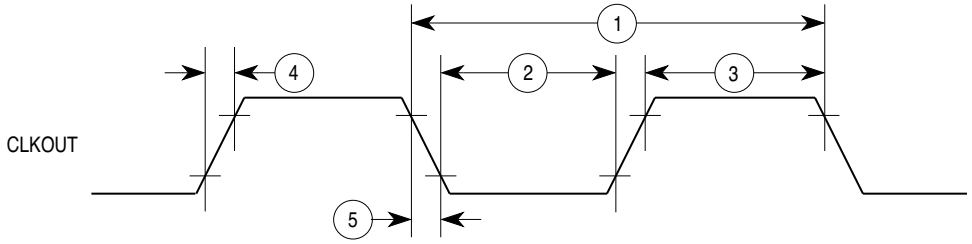
   $$V_{errj} = R_S \times I_{OFF}$$

   where I_{OFF} is a function of operating temperature. (See Table A–10, note 4).

   Charge-sharing leakage is a function of ADC clock speed, number of channels scanned, and source impedance:

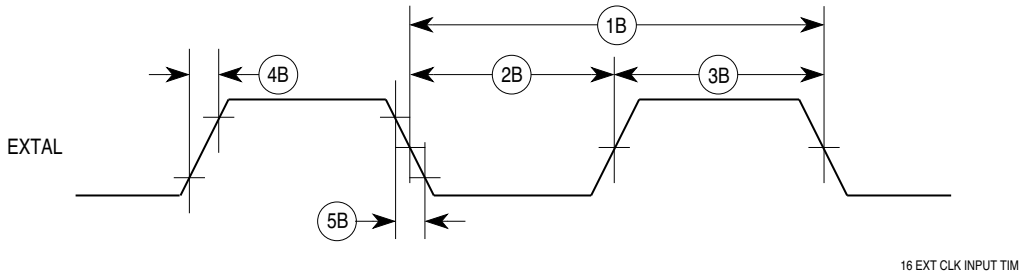
   For 10-bit conversion, $V_{err10} = .25\ pF \times V_{DDA} \times R_S \times ADCLK \div (9 \times \text{number of channels})$

   For 8-bit conversion, $V_{err8} = .25\ pF \times V_{DDA} \times R_S \times ADCLK \div (8 \times \text{number of channels})$



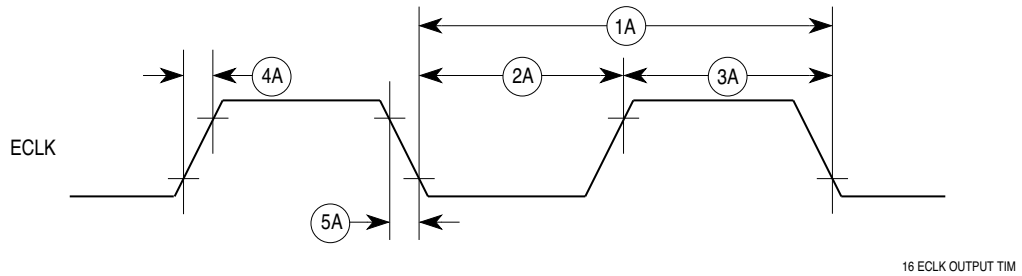
NOTE: Timing shown with respect to 20% and 70% V_{DD}.

Figure 21 CLKOUT Output Timing Diagram



NOTE: Timing shown with respect to 20% and 70% V_{DD}. Pulse width shown with respect to 50% V_{DD}.

Figure 22 External Clock Input Timing Diagram



NOTE: Timing shown with respect to 20% and 70% V_{DD}.

Figure 23 ECLK Output Timing Diagram

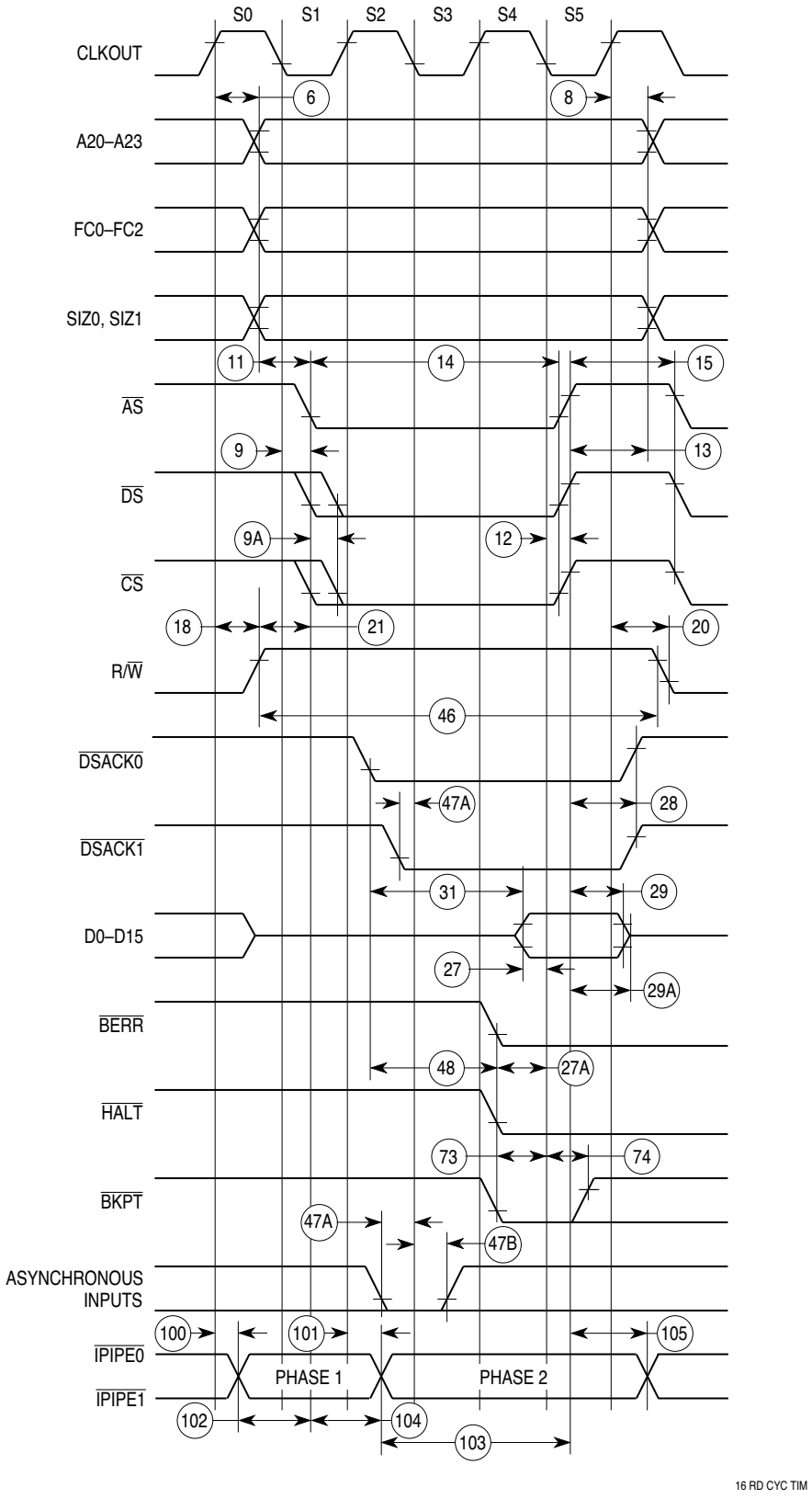


Figure 24 Read Cycle Timing Diagram

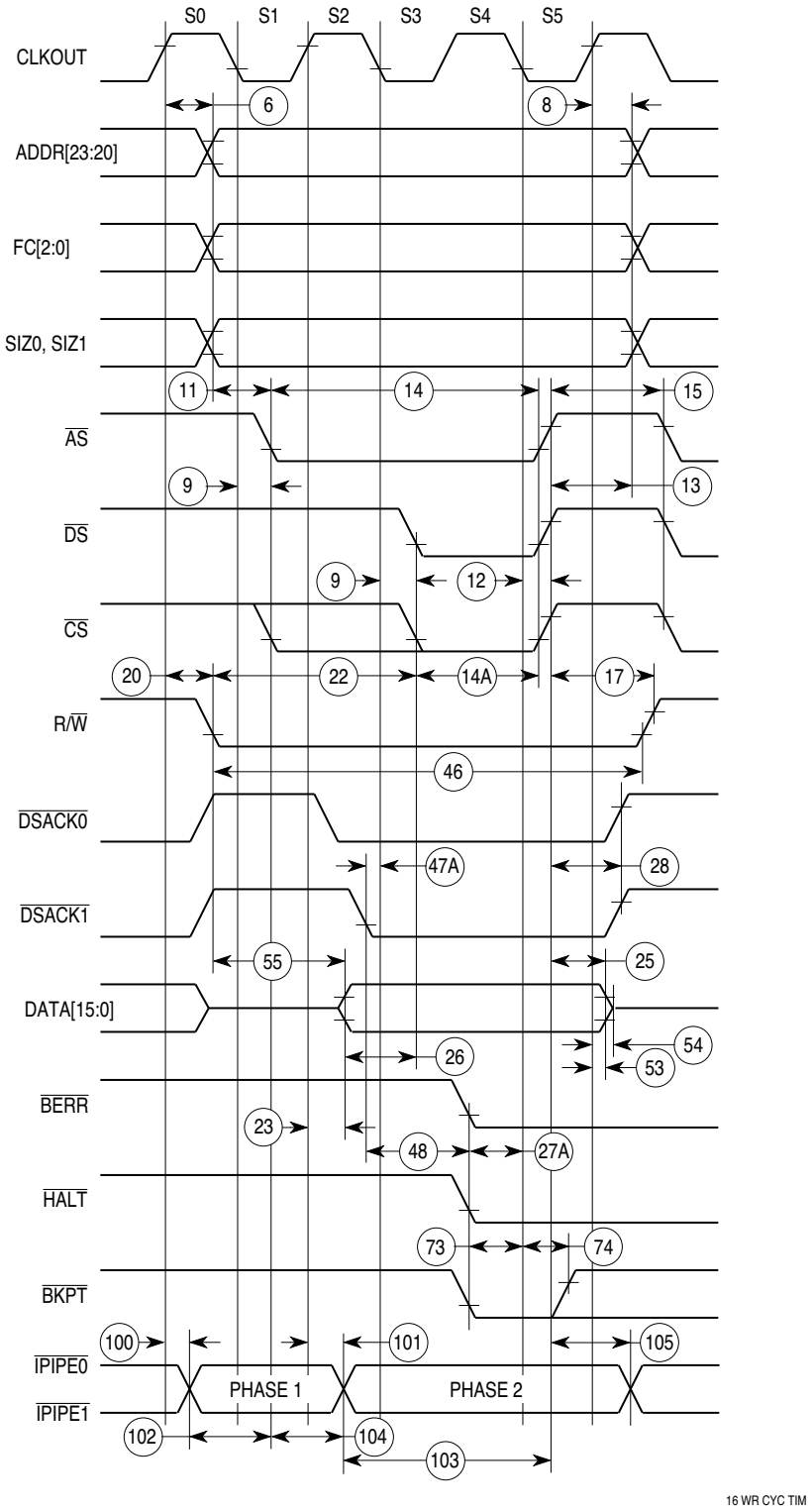


Figure 25 Write Cycle Timing Diagram

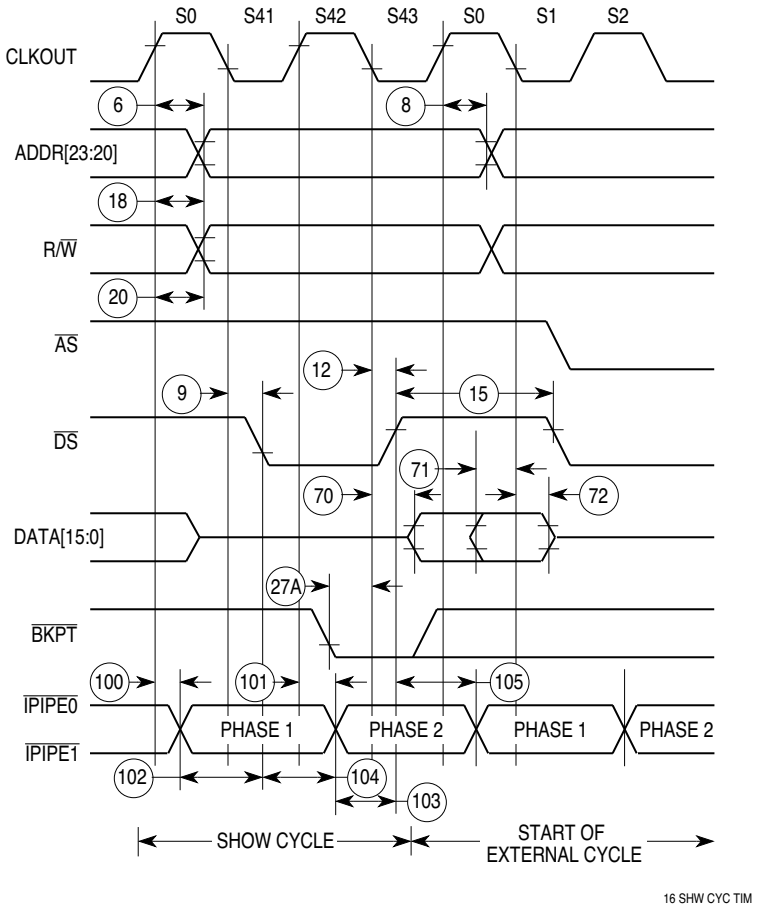


Figure 26 Show Cycle Timing Diagram

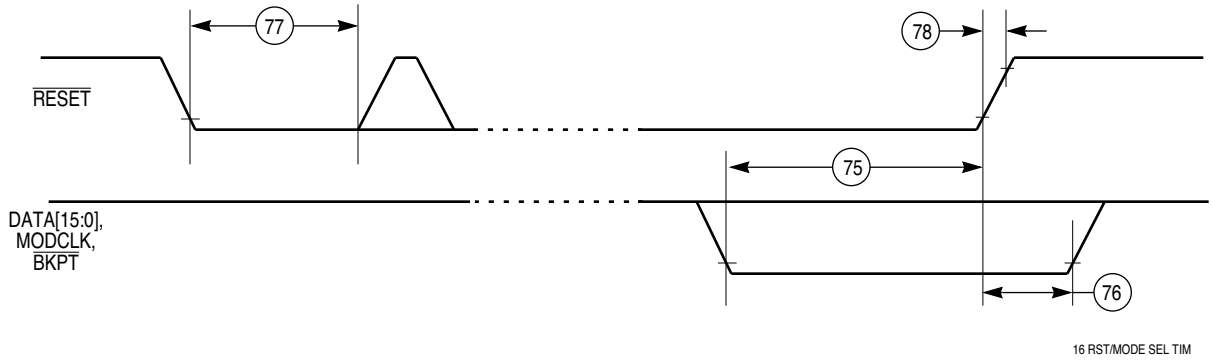


Figure 27 Data Bus Mode Select Timing Diagram

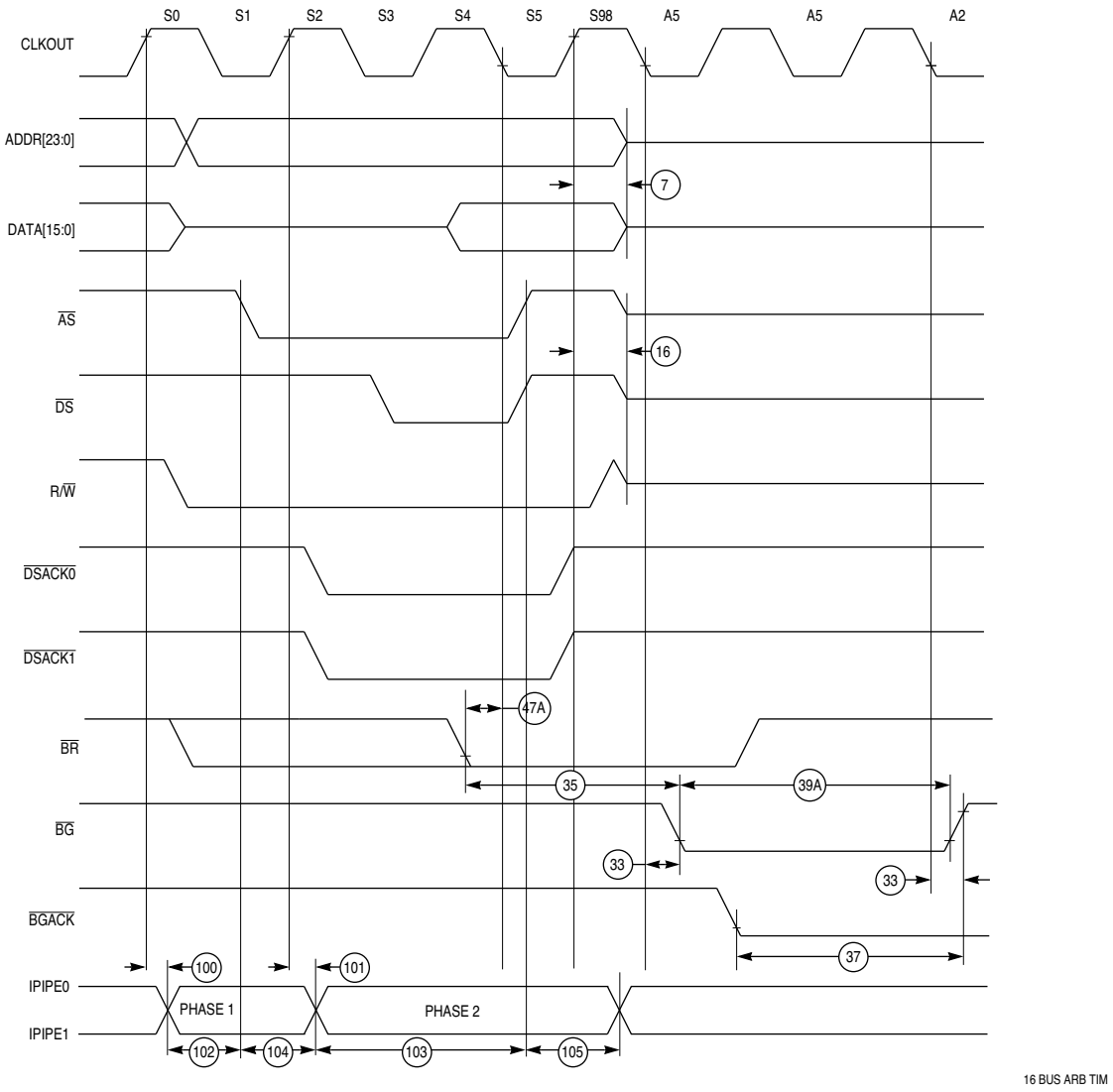


Figure 28 Bus Arbitration Timing Diagram — Active Bus Case

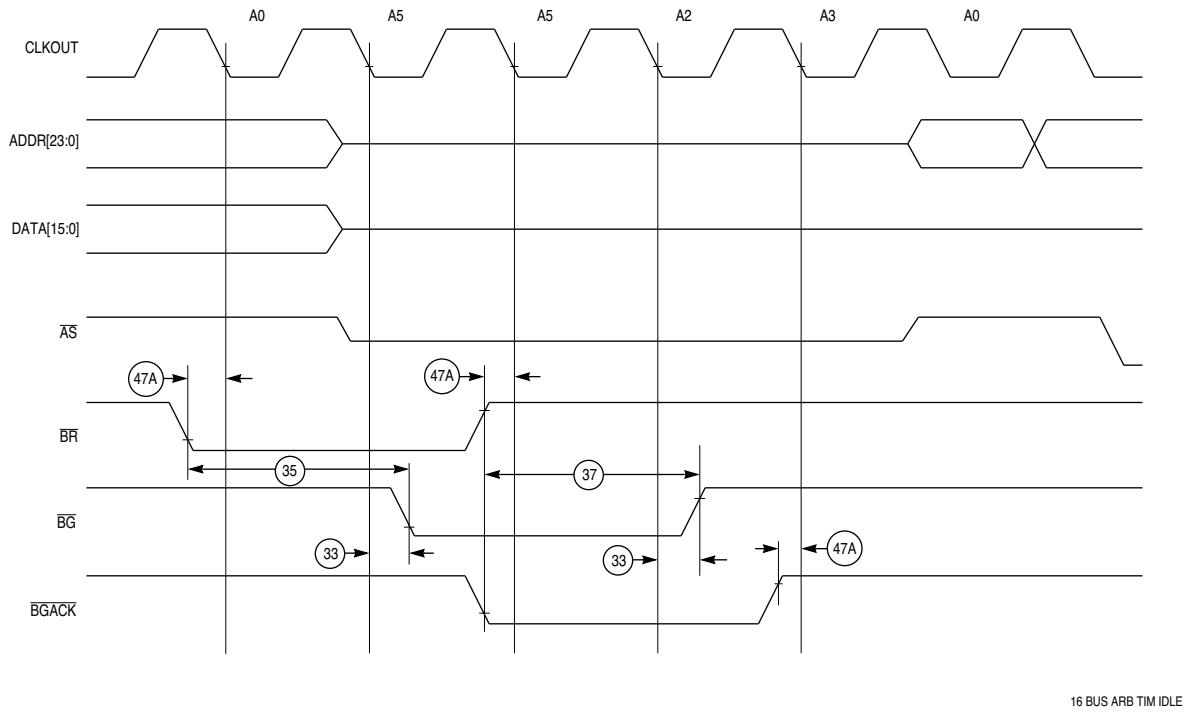


Figure 29 Bus Arbitration Timing Diagram — Idle Bus Case

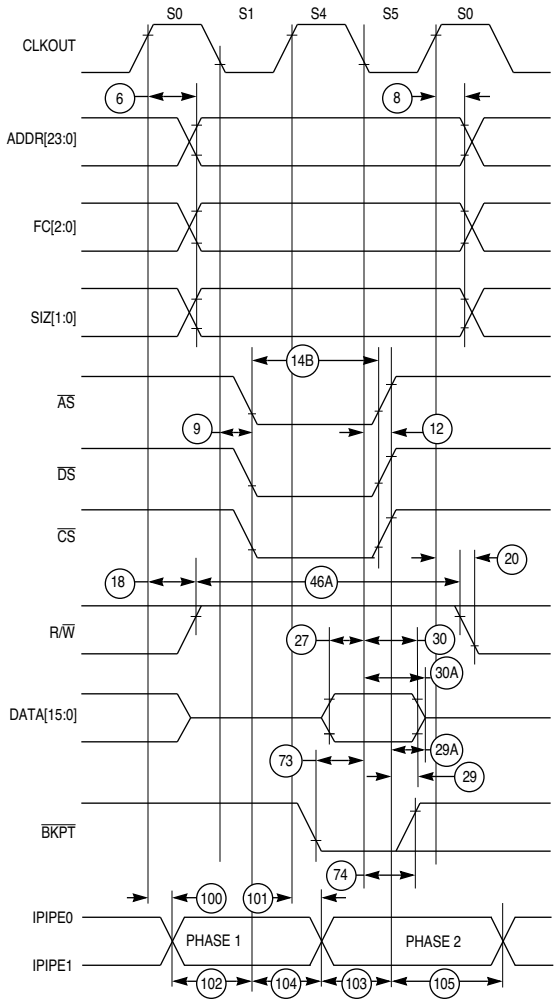


Figure 30 Fast Termination Read Cycle Timing Diagram

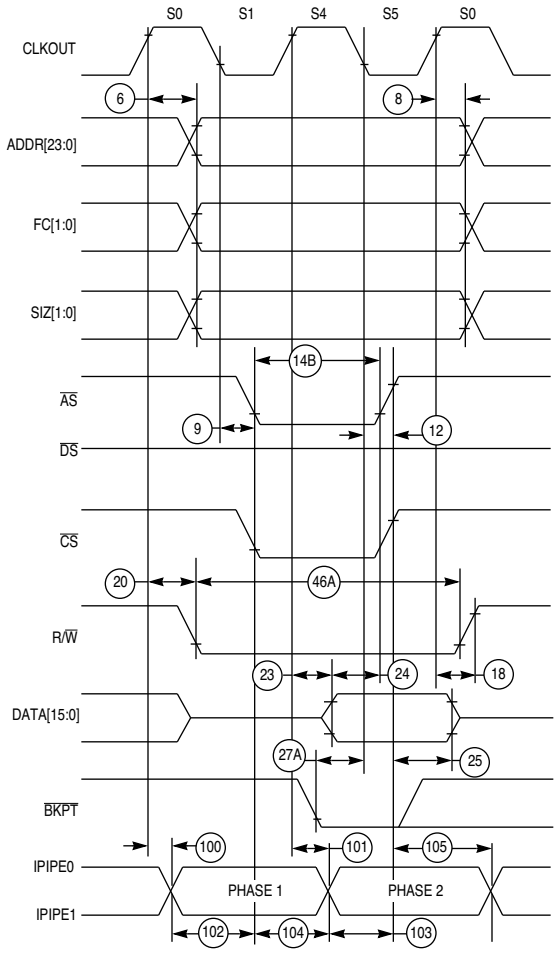
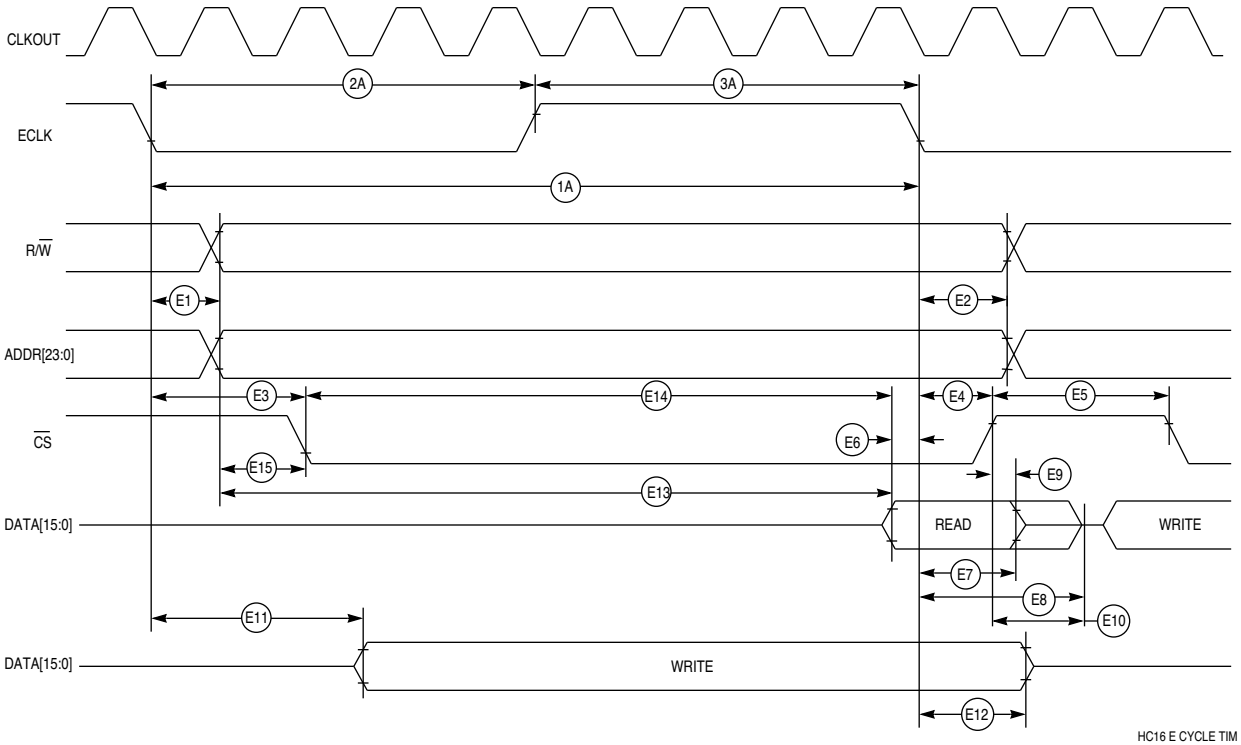
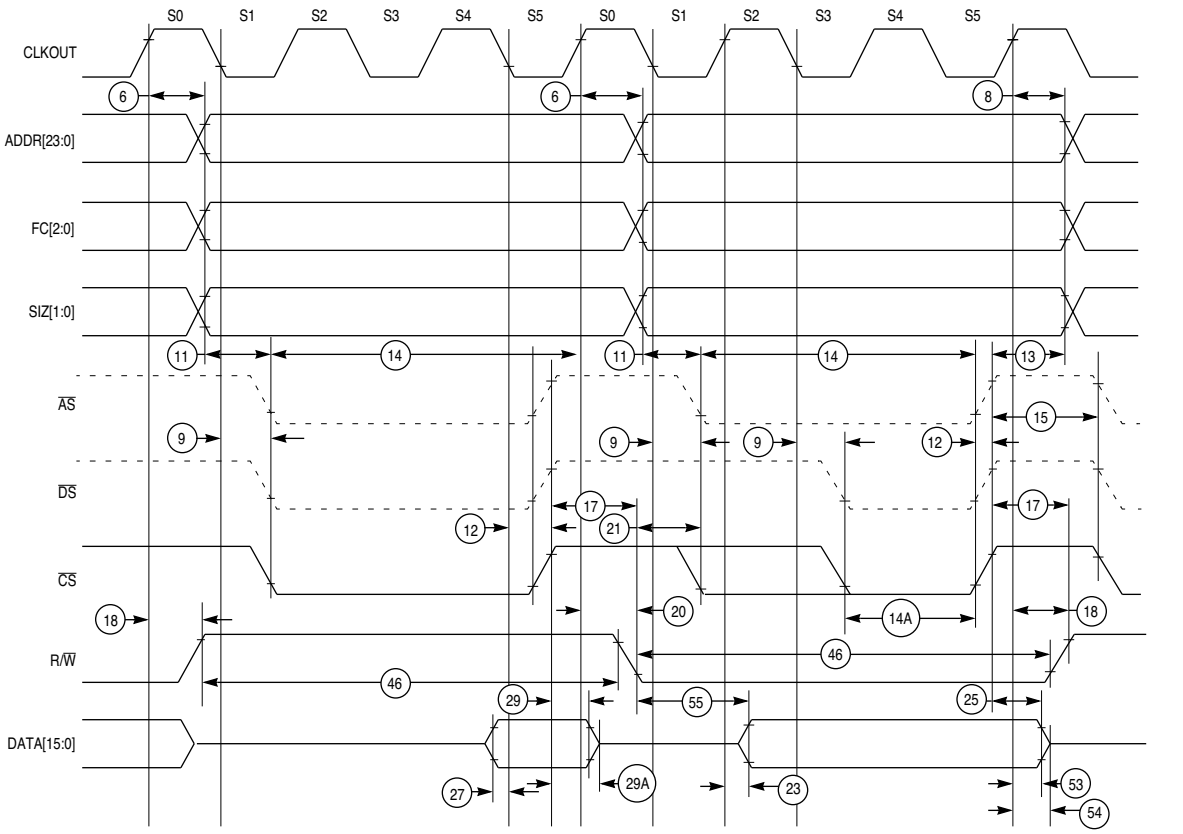


Figure 31 Fast Termination Write Cycle Timing Diagram



NOTE: Shown with ECLK = system clock/8 — EDIV bit in clock synthesizer control register (SYNCR) = 0.

Figure 32 ECLK Timing Diagram



NOTE: $\overline{AS}$ and $\overline{DS}$ timing shown for reference only.

Figure 33 Chip Select Timing Diagram

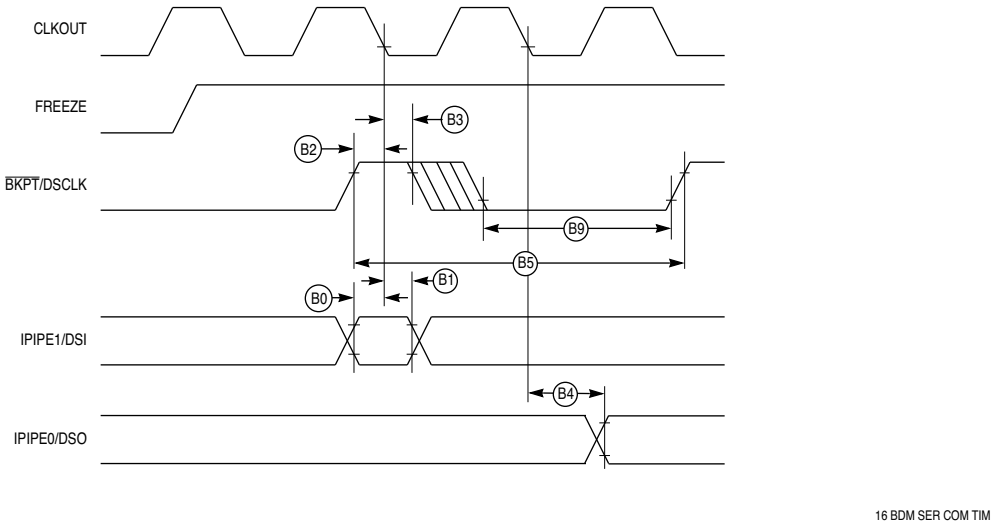


Figure 34 Background Debugging Mode Timing Diagram — Serial Communication

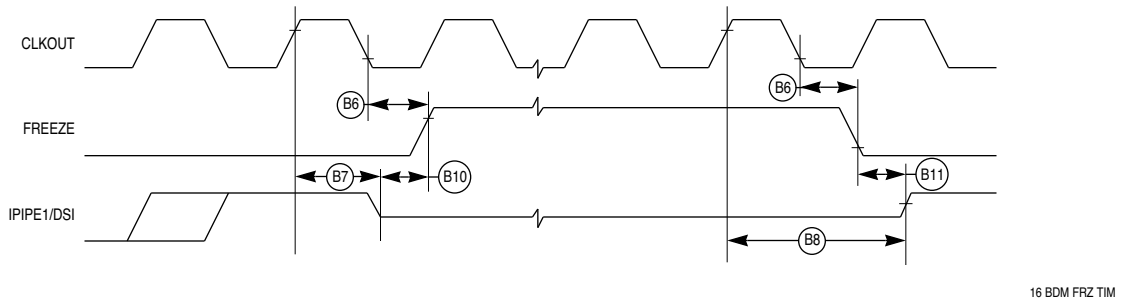


Figure 35 Background Debugging Mode Timing Diagram —Freeze Assertion

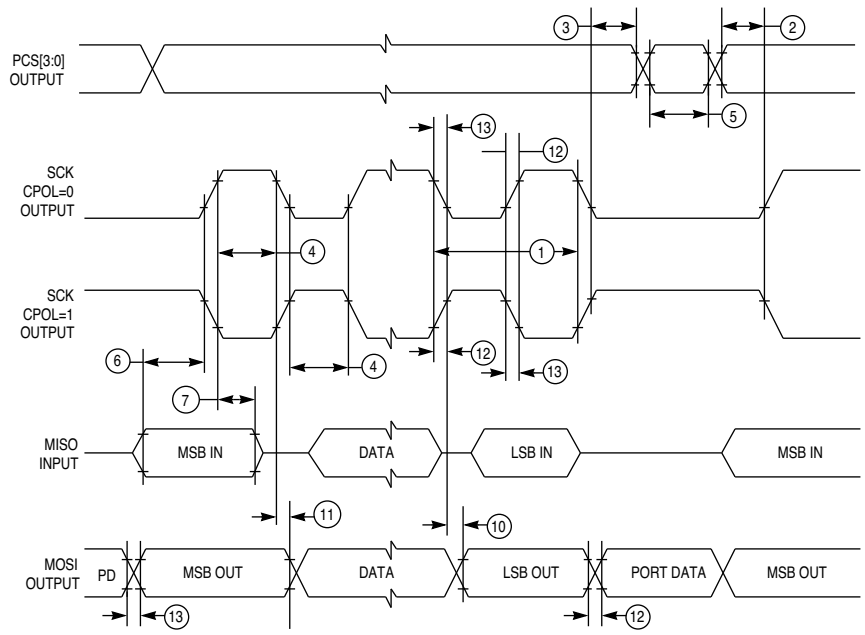


Figure 36 QSPI Timing Master, CPHA = 0

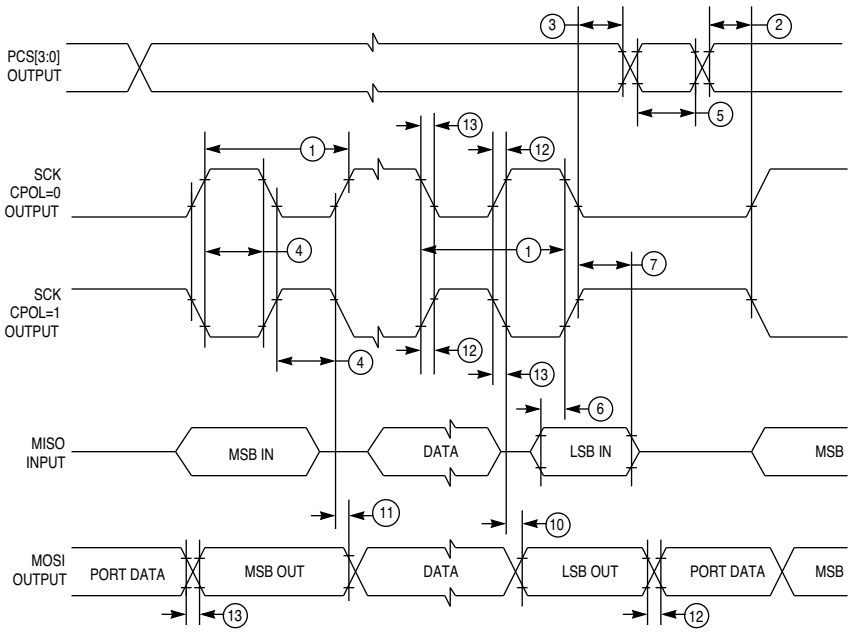
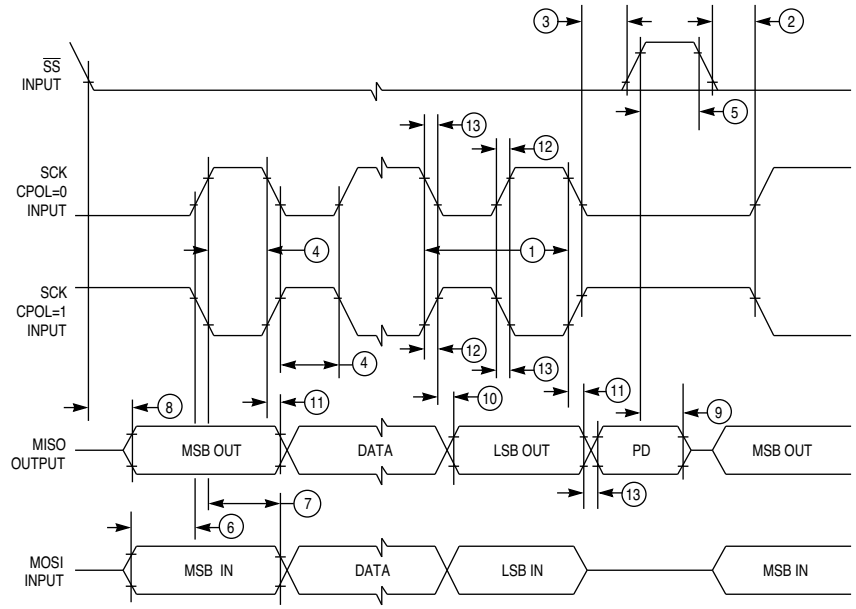
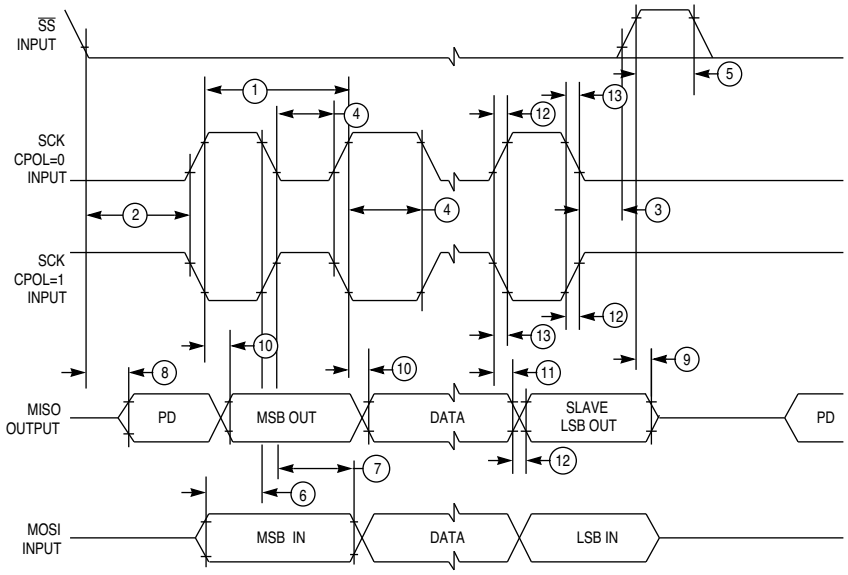


Figure 37 QSPI Timing Master, CPHA = 1



16 QSPI SLV CPHA0

Figure 38 QSPI Timing Slave, CPHA = 0



16 QSPI SLV CPHA1

Figure 39 QSPI Timing Slave, CPHA = 1

9 Summary of Changes

This is a partial revision. Most of the publication remains the same, but the following changes were made to improve it. Typographical errors that do not affect content are not annotated.

Page 2	Ordering information. All currently available options added.
Page 5	Block diagram revised. All pin functions shown, port mnemonics changed.
Page 6	Pinout diagram revised. All pin functions shown, port mnemonics changed.
Page 7	144-pin diagram added.
Pages 8-9	Corrected port assignments, new notes, changed B driver description.
Pages 10-12	Corrected port assignments, new notes, changed B driver description.
Pages 13-16	Revised register map, new pseudolinear maps.
Page 17	Added XMSK, YMSK registers to diagram.
Page 41	SIM memory map standardized.
Page 45	Expanded IARB field description.
Pages 48-49	Expanded system clock description.
Page 51	Expanded and relocated PIT description.
Page 65	New information concerning PE3.
Pages 66-69	New resets section.
Pages 70-72	New interrupts section.
Page 75	ADC memory map standardized, result register mnemonics added.
Pages 76 &77	Changed ADC I/O port register mnemonics to reflect port name.
Page 77	Changed prescaler rate selection values.
Page 83	QSM memory map standardized.
Pages 83 & 86	Changed QSM I/O port register mnemonics to reflect port name.
Page 94	New QSPI RAM diagram.
Pages 90 & 91	Changed SPI BR field mnemonic to SPBR.
Page 97	Changed SCI BR field mnemonic to SCBR.
Page 102	SRAM memory map added.
Page 106	GPT memory map standardized.
Pages 106 & 110	Changed GPT I/O port register mnemonics to reflect port name.
Pages 118-145	New electrical characteristics section.

NOTES

NOTES

NOTES

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

How to reach us:
USA/EUROPE: Motorola Literature Distribution;
P.O. Box 20912; Phoenix, Arizona 85036. 1-800-441-2447
MFAX: RMFAX0@email.sps.mot.com - TOUCHTONE (602) 244-6609
INTERNET: http://Design-NET.com
JAPAN: Nippon Motorola Ltd.; Tatsumi-SPD-JLDC, Toshikatsu Otsuki,
6F Seibu-Butsuryu-Center, 3-14-2 Tatsumi Koto-Ku, Tokyo 135, Japan. 03-3521-8315
HONG KONG: Motorola Semiconductors H.K. Ltd.; 8B Tai Ping Industrial Park,
51 Ting Kok Road, Tai Po, N.T., Hong Kong. 852-26629298

MC68HC16Z1TS/D