

# MC68HC812A4

## *Technical Summary* **16-Bit Microcontroller**

### **1 Introduction**

The MC68HC812A4 microcontroller unit (MCU) is a 16-bit device composed of standard on-chip peripheral modules connected by an intermodule bus. Modules include a 16-bit central processing unit (CPU12), a Lite integration module (LIM), two asynchronous serial communications interfaces (SCI0 and SCI1), a serial peripheral interface (SPI), a timer and pulse accumulation module, an 8-bit analog-to-digital converter (ATD), 1-Kbyte RAM, 4-Kbyte EEPROM, and memory expansion logic with chip selects, key wakeup ports, and a phase-locked loop (PLL).

#### **1.1 Features**

- Low-Power, High-Speed M68HC12 CPU
- Power Saving STOP and WAIT Modes
- Memory
  - 1024-Byte RAM
  - 4096-Byte Electrically Erasable Programmable Read-Only Memory (EEPROM)
  - On-Chip Memory Mapping Allows Expansion to over 5-Mbyte Address Space
- Single-Wire Background Debug Mode
- Non-Multiplexed Address and Data Buses
- Seven Programmable Chip Selects with Clock Stretching (Expanded Modes)
- 8-Channel, Enhanced 16-Bit Timer with Programmable Prescaler
  - All Channels Configurable as Input Capture or Output Compare
  - Flexible Choice of Clock Source
- 16-Bit Pulse Accumulator
- Real-Time Interrupt Circuit
- Computer Operating Properly (COP) Watchdog
- Clock Monitor
- Phase-Locked Loop
- Two Enhanced Asynchronous Non-Return to Zero (NRZ) Serial Communication Interfaces (SCI)
- Enhanced Synchronous Serial Peripheral Interface (SPI)
- 8-Channel, 8-Bit Analog-to-Digital Converter (ATD)
- Up to 24 Key Wakeup Lines with Interrupt Capability
- Available in 112-Pin Thin Quad Flat Pack (TQFP) Packaging

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



## 1.2 Ordering Information

**Table 1 MC68HC812A4 Device Ordering Information**

Package	Temperature		Voltage	Frequency	Order Number
	Range	Designator			
112-Pin TQFP Single Tray 60 Pcs	0 to +70C		4.5V–5.5V	8 MHz	MC68HC812A4PV8
	-40 to +85C	C			MC68HC812A4CPV8
	-40 to +105C	V			MC68HC812A4VPV8
	-40 to +125C	M			MC68HC812A4MPV8

NOTE: This part is also available in 2-piece sample packs and 300-piece bricks.

Evaluation boards, assemblers, compilers, and debuggers are available from Motorola and from third-party suppliers. An up-to-date list of products that support the M68HC12 family of microcontrollers can be found on the World Wide Web at the following URL:

<http://www.mcu.mot.sps.com>

Documents to assist in product selection are available from the Motorola Literature Distribution Center or your local Motorola Sales Office:

AMCU Device Selection Guide (SG166/D)

AMCU Software and Development Tool Selector Guide (SG176/D)

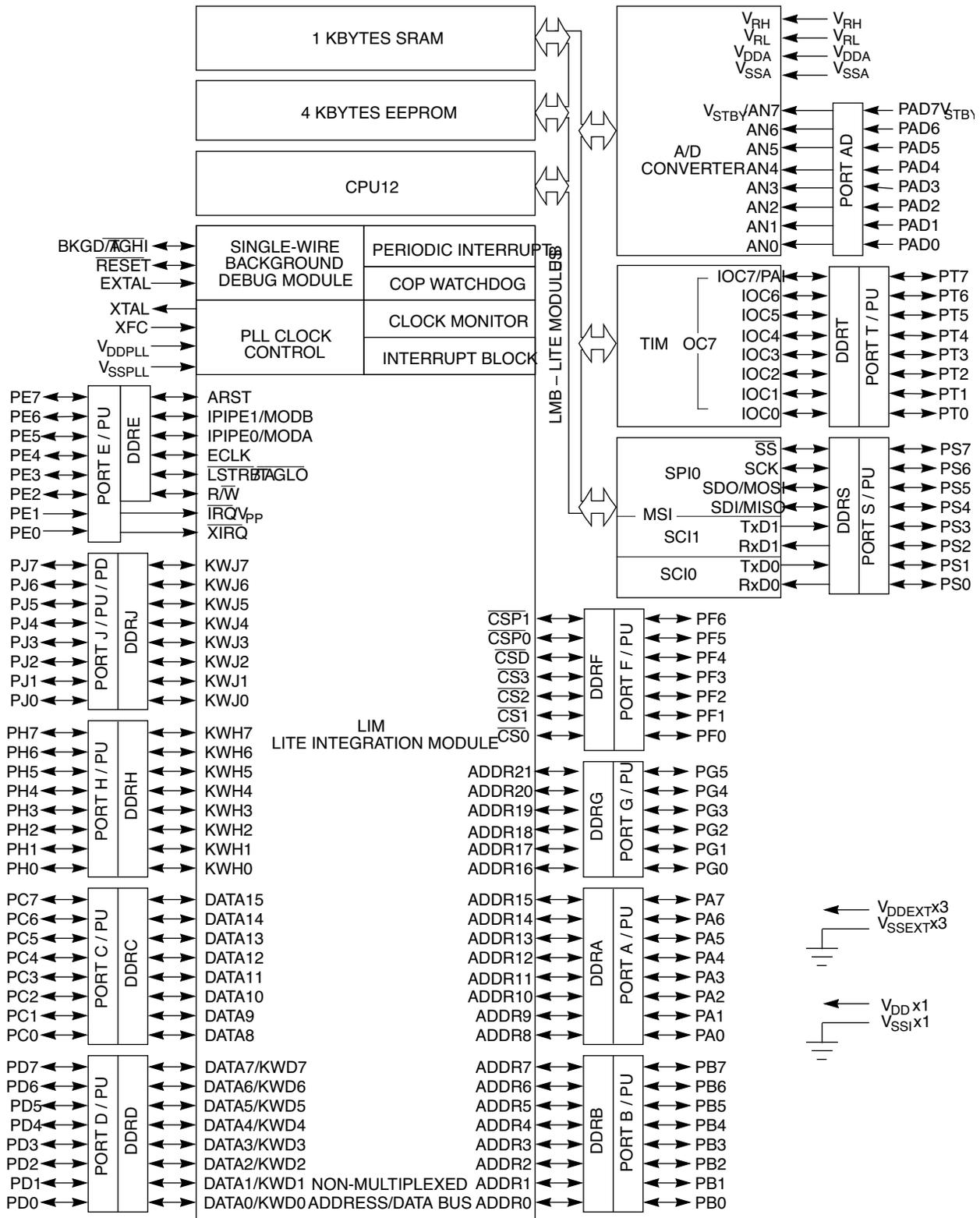
# TABLE OF CONTENTS

Section		Page
<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Features .....	1
1.2	Ordering Information .....	2
1.3	MC68HC812A4 Block Diagram .....	5
<b>2</b>	<b>Central Processing Unit</b>	<b>6</b>
2.1	Programming Model .....	6
2.2	Data Types .....	7
2.3	Addressing Modes .....	7
2.4	Indexed Addressing Modes .....	8
2.5	Opcodes and Operands .....	8
<b>3</b>	<b>Pinout and Signal Descriptions</b>	<b>9</b>
3.1	MC68HC812A4 Pin Assignments .....	9
3.2	Signal Descriptions .....	10
<b>4</b>	<b>Register Block</b>	<b>13</b>
<b>5</b>	<b>Bus Control and Input/Output</b>	<b>17</b>
5.1	Detecting Access Type from External Signals .....	17
5.2	Registers .....	17
<b>6</b>	<b>Operating Modes and Resource Mapping</b>	<b>25</b>
6.1	Operating Modes .....	25
6.2	Background Debug Mode .....	26
6.3	Internal Resource Mapping .....	28
6.4	Memory Maps .....	31
<b>7</b>	<b>EEPROM</b>	<b>32</b>
7.1	EEPROM Programmer's Model .....	32
7.2	EEPROM Control Registers .....	33
<b>8</b>	<b>Memory Expansion and Chip Select</b>	<b>37</b>
8.1	General Description of Memory Expansion .....	37
8.2	Generation of Chip Selects .....	37
8.3	Chip Select Stretch .....	41
8.4	Memory Expansion Registers .....	43
8.5	Chip Selects .....	45
8.6	Priority .....	47
<b>9</b>	<b>Resets and Interrupts</b>	<b>48</b>
9.1	Exception Priority .....	48
9.2	Maskable interrupts .....	48
9.3	Interrupt Control and Priority Registers .....	49
9.4	Resets .....	50
9.5	Effects of Reset .....	51
9.6	Register Stacking .....	52
<b>10</b>	<b>Key Wakeups</b>	<b>53</b>
10.1	Key Wakeup Registers .....	53
<b>11</b>	<b>Clock Functions</b>	<b>57</b>
11.1	Clock Sources .....	57
11.2	Computer Operating Properly (COP) .....	57
11.3	Real-Time Interrupt .....	57
11.4	Clock Monitor .....	57
11.5	Clock Function Registers .....	58
11.6	Clock Divider Chains .....	61
<b>12</b>	<b>Phase-Locked Loop</b>	<b>63</b>
12.1	PLL Register Description .....	64

## TABLE OF CONTENTS (Continued)

Section	Page
<b>13 Standard Timer Module</b>	<b>66</b>
13.1 Timer Registers .....	67
13.2 Timer Operation in Modes .....	75
<b>14 Multiple Serial Interface</b>	<b>76</b>
14.1 Block diagram .....	76
14.2 Serial Communication Interface (SCI) .....	76
14.3 Serial Peripheral Interface (SPI) .....	83
14.4 Port S .....	89
<b>15 Analog-to-Digital Converter</b>	<b>91</b>
15.1 Functional Description .....	91
15.2 ATD Registers .....	92
15.3 ATD Mode Operation .....	97
<b>16 Development Support</b>	<b>98</b>
16.1 Instruction Queue .....	98
16.2 Background Debug Mode .....	98
16.3 Instruction Tagging .....	106
<b>17 Summary of Changes</b>	<b>107</b>

### 1.3 MC68HC812A4 Block Diagram



HC812A4 BLOCK

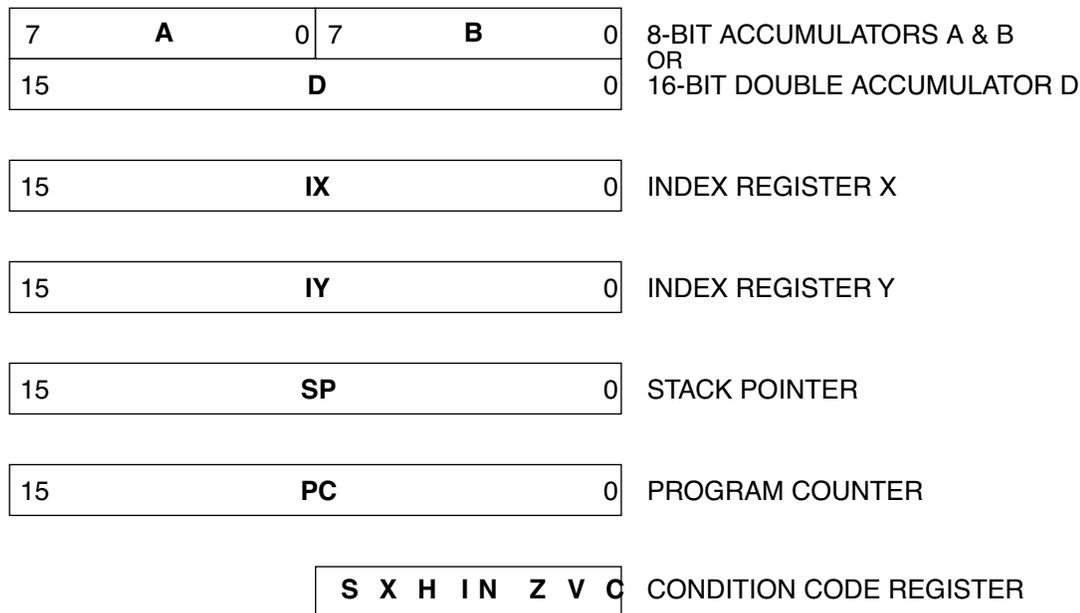
Figure 1 MC68HC812A4 Block Diagram

## 2 Central Processing Unit

The CPU12 is a high-speed, 16-bit processing unit. It has full 16-bit data paths and wider internal registers (up to 20 bits) for high-speed extended math instructions. The instruction set is a proper superset of the M68HC11 instruction set. The CPU12 allows instructions with odd byte counts, including many single-byte instructions. This provides efficient use of ROM space. An instruction queue buffers program information so the CPU always has immediate access to at least three bytes of machine code at the start of every instruction. The CPU12 also offers an extensive set of indexed addressing capabilities.

### 2.1 Programming Model

CPU12 registers are an integral part of the CPU and are not addressed as if they were memory locations.



HC12 PROG MODEL

**Figure 2 Programming Model**

**Accumulators** A and B are general-purpose 8-bit accumulators used to hold operands and results of arithmetic calculations or data manipulations. Some instructions treat the combination of these two 8-bit accumulators as a 16-bit double accumulator (accumulator D).

**Index registers** X and Y are used for indexed addressing mode. In the indexed addressing mode, the contents of a 16-bit index register are added to 5-bit, 9-bit, or 16-bit constants or the content of an accumulator to form the effective address of the operand to be used in the instruction.

**Stack pointer** (SP) points to the last stack location used. The CPU12 supports an automatic program stack that is used to save system context during subroutine calls and interrupts, and can also be used for temporary storage of data. The stack pointer can also be used in all indexed addressing modes.

**Program counter** is a 16-bit register that holds the address of the next instruction to be executed. The program counter can be used in all indexed addressing modes except autoincrement/decrement.

**Condition code register** (CCR) contains five status indicators, two interrupt masking bits, and a STOP disable bit. The five flags are half carry (H), negative (N), zero (Z), overflow (V), and carry/borrow (C). The half-carry flag is used only for BCD arithmetic operations. The N, Z, V, and C status bits allow for branching based on the results of a previous operation.

## 2.2 Data Types

The CPU12 supports the following data types:

- Bit data
- 8-bit and 16-bit signed and unsigned integers
- 16-bit unsigned fractions
- 16-bit addresses

A byte is eight bits wide and can be accessed at any byte location. A word is composed of two consecutive bytes with the most significant byte at the lower value address. There are no special requirements for alignment of instructions or operands.

## 2.3 Addressing Modes

Addressing modes determine how the CPU accesses memory locations to be operated upon. The CPU12 includes all of the addressing modes of the M68HC11 CPU as well as several new forms of indexed addressing. **Table 2** is a summary of the available addressing modes.

**Table 2 M68HC12 Addressing Mode Summary**

Addressing Mode	Source Format	Abbreviation	Description
Inherent	INST (no externally supplied operands)	INH	Operands (if any) are in CPU registers
Immediate	INST #opr8i or INST #opr16i	IMM	Operand is included in instruction stream 8- or 16-bit size implied by context
Direct	INST opr8a	DIR	Operand is the lower 8-bits of an address in the range \$0000 – \$00FF
Extended	INST opr16a	EXT	Operand is a 16-bit address
Relative	INST rel8 or INST rel16	REL	An 8-bit or 16-bit relative offset from the current pc is supplied in the instruction
Indexed (5-bit offset)	INST oprx5,xysp	IDX	5-bit signed constant offset from x, y, sp, or pc
Indexed (auto pre-decrement)	INST oprx3,-xys	IDX	Auto pre-decrement x, y, or sp by 1 ~ 8
Indexed (auto pre-increment)	INST oprx3,+xys	IDX	Auto pre-increment x, y, or sp by 1 ~ 8
Indexed (auto post-decrement)	INST oprx3,xys-	IDX	Auto post-decrement x, y, or sp by 1 ~ 8
Indexed (auto post-increment)	INST oprx3,xys+	IDX	Auto post-increment x, y, or sp by 1 ~ 8
Indexed (accumulator offset)	INST abd,xysp	IDX	Indexed with 8-bit (A or B) or 16-bit (D) accumulator offset from x, y, sp, or pc
Indexed (9-bit offset)	INST oprx9,xysp	IDX1	9-bit signed constant offset from x, y, sp, or pc (lower 8-bits of offset in one extension byte)
Indexed (16-bit offset)	INST oprx16,xysp	IDX2	16-bit constant offset from x, y, sp, or pc (16-bit offset in two extension bytes)
Indexed-Indirect (16-bit offset)	INST [oprx16,xysp]	[IDX2]	Pointer to operand is found at... 16-bit constant offset from x, y, sp, or pc (16-bit offset in two extension bytes)
Indexed-Indirect (D accumulator offset)	INST [D,xysp]	[D,IDX]	Pointer to operand is found at... x, y, sp, or pc plus the value in D

## 2.4 Indexed Addressing Modes

The CPU12 indexed modes reduce execution time and eliminate code size penalties for using the Y index register. CPU12 indexed addressing uses a postbyte plus zero, one, or two extension bytes after the instruction opcode. The postbyte and extensions do the following tasks:

- Specify which index register is used.
- Determine whether a value in an accumulator is used as an offset.
- Enable automatic pre- or post-increment or decrement
- Specify use of 5-bit, 9-bit, 9-bit, or 16-bit signed offsets.

**Table 3 Summary of Indexed Operations**

Postbyte Code (xb)	Source Code Syntax	Comments
rr0nnnnn	,r n,r -n,r	<b>5-bit constant offset</b> n = -16 to +15 rr can specify X, Y, SP, or PC
111rr0zs	n,r -n,r	<b>Constant offset</b> (9- or 16-bit signed) z- 0 = 9-bit with sign in LSB of postbyte(s) 1 = 16-bit if z = s = 1, 16-bit offset indexed-indirect (see below) rr can specify X, Y, SP, or PC
111rr011	[n,r]	<b>16-bit offset indexed-indirect</b> rr can specify X, Y, SP, or PC
rr1pnnnn	n,-r n,+r n,r- n,r+	<b>Auto pre-decrement/increment or Auto post-decrement/increment;</b> p = pre-(0) or post-(1), n = -8 to -1, +1 to +8 rr can specify X, Y, or SP (PC not a valid choice)
111rr1aa	A,r B,r D,r	<b>Accumulator offset</b> (unsigned 8-bit or 16-bit) aa- 00 = A 01 = B 10 = D (16-bit) 11 = see accumulator D offset indexed-indirect rr can specify X, Y, SP, or PC
111rr111	[D,r]	<b>Accumulator D offset indexed-indirect</b> rr can specify X, Y, SP, or PC

## 2.5 Opcodes and Operands

The CPU12 uses 8-bit opcodes. Each opcode identifies a particular instruction and associated addressing mode to the CPU. Several opcodes are required to provide each instruction with a range of addressing capabilities.

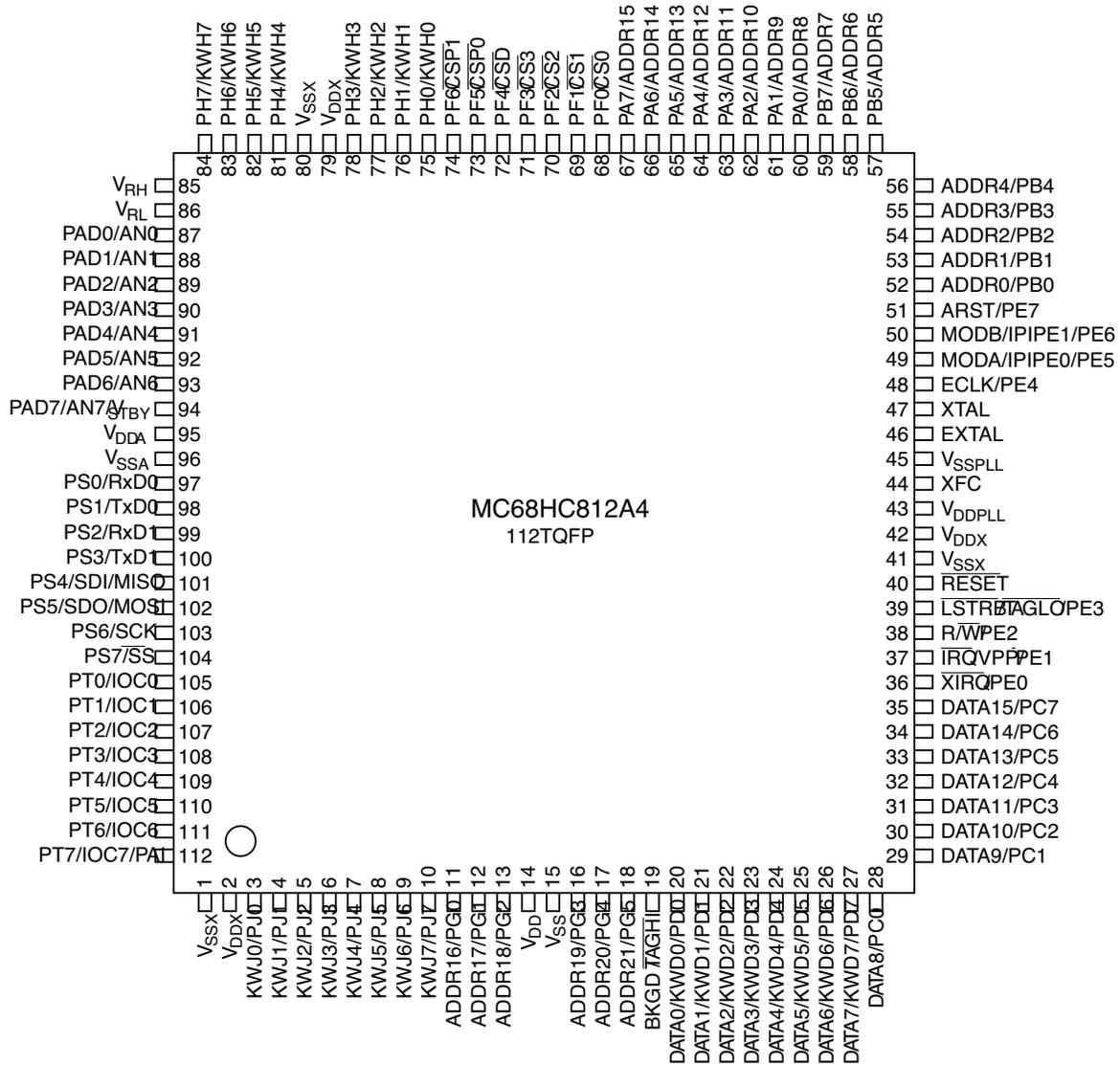
Only 256 opcodes would be available if the range of values were restricted to the number that can be represented by 8-bit binary numbers. To expand the number of opcodes, a second page is added to the opcode map. Opcodes on the second page are preceded by an additional byte with the value \$18.

To provide additional addressing flexibility, opcodes can also be followed by a postbyte or extension bytes. Postbytes implement certain forms of indexed addressing, transfers, exchanges, and loop primitives. Extension bytes contain additional program information such as addresses, offsets, and immediate data.

### 3 Pinout and Signal Descriptions

#### 3.1 MC68HC812A4 Pin Assignments

The MC68HC812A4 is available in a 112-pin thin quad flat pack (TQFP). Most pins perform two or more functions, as described in the **3.2 Signal Descriptions**. **Figure 3** shows pin assignments.



HC12 112TQFP

Figure 3 Pin Assignments for MC68HC812A4

### 3.2 Signal Descriptions

MC68HC812A4 pins and signals are described in **Table 4**. Individual ports are cross referenced in **Table 5**.

**Table 4 MC68HC812A4 Signal Descriptions**

Mnemonic	Port	Description
V <sub>DD</sub> , V <sub>SS</sub>	—	V <sub>DD</sub> is the power supply, and V <sub>SS</sub> is ground. The MCU operates from a single power supply. Use customary bypass techniques as very fast signal transitions occur on the MCU pins.
V <sub>RH</sub> , V <sub>RL</sub>	—	Provide the reference voltage for the analog-to-digital converter.
AV <sub>DD</sub> , AV <sub>SS</sub>	—	Provides the operating voltage and ground for the analog-to-digital converter. This allows the supply voltage to the ATD to be bypassed independently.
V <sub>DDPLL</sub> , V <sub>SSPLL</sub>	—	Power and ground for PLL clock control; allows independent supply voltage to the PLL.
V <sub>STBY</sub>	Port AD	V <sub>STBY</sub> is used to input RAM standby power.
XTAL, EXTAL	—	Crystal driver and external clock input pins provide the interface for either a crystal or a CMOS compatible clock to control the internal clock generator circuitry. Out of reset the frequency applied to EXTAL is two times higher than the desired E-clock rate. On reset all the device clocks are derived from the EXTAL input frequency. XTAL is the crystal output.
XIRQ	PE0	Provides a means of requesting a non-maskable interrupt request after reset initialization.
IRQ	PE1	Maskable interrupt request input provides a means of applying asynchronous interrupt requests to the MCU. Either falling edge-sensitive triggering or level-sensitive triggering is program selectable (INTCR register).
R/W	PE2	Indicates direction of data on expansion bus. Shares function with general-purpose I/O. Read/write in expanded modes.
LSTRB	PE3	Low byte strobe (0 = low byte valid), in all modes this pin can be used as I/O. The low strobe function is the exclusive-NOR of A0 and the internal SZ8 signal. (The SZ8 internal signal indicates the size 16/8 access.)
ECLK	PE4	E-clock is the output connection for the external bus clock. ECLK is used as a timing reference. The unstretched ECLK frequency is normally equal to 1/2 the crystal frequency. Can be general-purpose I/O.
BKGD	—	State of mode select pins during reset determine the initial operating mode of the MCU. After reset, MODA and MODB can be configured as instruction queue tracking signals IPIPE0 and IPIPE1 or as general-purpose I/O pins.
MODA	PE5	Instruction queue tracking signals can be used by a development system to reconstruct the instruction queue and track instruction execution.
MODB	PE6	
IPIPE0	PE5	Instruction queue tracking signals can be used by a development system to reconstruct the instruction queue and track instruction execution.
IPIPE1	PE6	
ARST	PE7	Alternate reset input or general-purpose I/O. It can be used as a separate controlled active-high reset input.
XFC	—	Loop filter pin for controlled damping of the PLL VCO loop.
RESET	—	An active low bidirectional control signal, RESET acts as an input to initialize the MCU to a known start-up state, and an output when COP or clock monitor causes a reset.
ADDR[15:8]	Port A	External bus pins share function with general-purpose I/O ports A, B, C, and D. In single-chip operating modes, the pins can be used for I/O; in expanded modes, the pins are used for the external buses. In narrow data bus mode, port D is available as standard I/O or key wakeup inputs.
ADDR[7:0]	Port B	
DATA[15:8]	Port C	
DATA[7:0]	Port D	
ADDR[21:16]	Port G	Memory expansion and general-purpose I/O.
CS[3:0] CSD CSP[1:0]	Port F	Chip selects and general-purpose I/O.

**Table 4 MC68HC812A4 Signal Descriptions (Continued)**

Mnemonic	Port	Description
BKGD	—	Single-wire background interface pin is dedicated to the background debug function. During reset, this pin determines special or normal operating mode.
KWD[7:0]	Port D	Key wakeup and general-purpose I/O; can cause an interrupt when an input transitions from high to low.
KWH[7:0]	Port H	
KWJ[7:0]	Port J	Key wakeup and general-purpose I/O; can cause an interrupt when an input transitions from high to low or from low to high.
RxD0	PS0	Serial communications interface receive pin for SCI0.
TxD0	PS1	Serial communications interface transmit pin for SCI0.
RxD1	PS2	Serial communications interface receive pin for SCI1.
TxD1	PS3	Serial communications interface transmit pin for SCI1.
SDI/MISO	PS4	Master in/slave out pin for serial peripheral interface.
SDO/MOSI	PS5	Master out/slave in pin for serial peripheral interface.
SCK	PS6	Serial clock for SPI system.
$\overline{SS}$	PS7	Slave select output for SPI master mode, input for slave mode.
IOC[7:0]	Port T	Input capture or output compare channels and pulse accumulator input.

**Table 5 Port Descriptions**

Port Name	Direction	Function
Port A	In/Out	General-purpose I/O in single-chip modes. External address bus ADDR[15:8] in expanded modes.
Port B	In/Out	General-purpose I/O in single-chip modes. External address bus ADDR[7:0] in expanded modes.
Port C	In/Out	General-purpose I/O in single-chip modes. External data bus DATA[15:8] in expanded wide modes; external data bus DATA[15:8]/DATA[7:0] in expanded narrow modes.
Port D	In/Out	General-purpose I/O in single-chip modes and expanded narrow modes. External data bus DATA[7:0] in expanded wide mode. As key wakeup can cause an interrupt when an input transitions from high to low.
Port E	PE[1:0] In PE[7:2] In/Out	Mode selection, bus control signals and interrupt service request signals; or general-purpose I/O.
Port F	In/Out	Chip select and general-purpose I/O.
Port G	In/Out	Memory expansion and general-purpose I/O.
Port H	In/Out	Key wakeup and general-purpose I/O, can cause an interrupt when an input transitions from high to low.
Port J	In/Out	Key wakeup and general-purpose I/O, can cause an interrupt when an input transitions from high to low or from low to high.
Port S	In/Out	Serial communications interface and serial peripheral interface subsystems and general-purpose I/O.
Port T	In/Out	Timer system and general-purpose I/O.
Port AD	In	Analog-to-digital converter and general-purpose input.

**Table 6 Port Pull-Up, Pulldown and Reduced Drive Summary**

Port Name	Resistive Input Loads	Enable Bit			Reduced Drive Control Bit		
		Register (Address)	Bit Name	Reset State	Register (Address)	Bit Name	Reset State
Port A	Pull-up	PUCR (\$000C)	PUPA	Enabled	RDRIV (\$000D)	RDPAB	Full Drive
Port B	Pull-up	PUCR (\$000C)	PUPB	Enabled	RDRIV (\$000D)	RDPAB	Full Drive
Port C	Pull-up	PUCR (\$000C)	PUPC	Enabled	RDRIV (\$000D)	RDPC	Full Drive
Port D	Pull-up	PUCR (\$000C)	PUPD	Enabled	RDRIV (\$000D)	RDPD	Full Drive
Port E:							
PE7, PE3, PE2, PE0	Pull-up	PUCR (\$000C)	PUPE	Enabled	RDRIV (\$000D)	RDPE	Full Drive
PE1	Pull-up	Always Enabled			RDRIV (\$000D)	RDPE	Full Drive
PE[6:4]	None	—			RDRIV (\$000D)	RDPE	Full Drive
PE[6:5]	Pulldown	Enabled During Reset			—	—	—
Port F	Pull-up	PUCR (\$000C)	PUPF	Enabled	RDRIV (\$000D)	RDPF	Full Drive
Port G	Pull-up	PUCR (\$000C)	PUPG	Enabled	RDRIV (\$000D)	RDPG	Full Drive
Port H	Pull-up	PUCR (\$000C)	PUPH	Enabled	RDRIV (\$000D)	RDPH	Full Drive
Port J	Pull-up/down <sup>1</sup>	PULEJ (\$002E)	PULEJ[7:0]	Disabled	RDRIV (\$000D)	RDPJ	Full Drive
Port S	Pull-up	SP0CR2 (\$00D1)	PUPS	Enabled	SP0CR2 (\$00D1)	RDS	Full Drive
Port T	Pull-up	TMSK2 (\$008D)	PUPT	Disabled	TMSK2 (\$008D)	RDPT	Full Drive
Port AD	None	—			—		
BKGD	Pull-up	—	—	Enabled	—	—	Full Drive

1. Pull-up or pulldown devices for each port J pin can be selected with the PUPSJ register (\$002D). After reset, pulldowns are selected for all port J pins but must be enabled with PULEJ register.

## 4 Register Block

The register block can be mapped to any 2-Kbyte boundary within the standard 64-Kbyte address space by manipulating bits REG[15:11] in the INITRG register. INITRG establishes the upper five bits of the register block's 16-bit address. The register block occupies the first 512 bytes of the 2-Kbyte block. Default addressing (after reset) is indicated in the table below. For additional information refer to **6 Operating Modes and Resource Mapping**.

**Table 7 MC68HC812A4 Register Map (Sheet 1 of 4)**

Address	Bit 7	6	5	4	3	2	1	Bit 0	Name
\$0000	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0	PORTA <sup>1</sup>
\$0001	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0	PORTB <sup>1</sup>
\$0002	Bit 7	6	5	4	3	2	1	Bit 0	DDRA <sup>1</sup>
\$0003	Bit 7	6	5	4	3	2	1	Bit 0	DDRB <sup>1</sup>
\$0004	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0	PORTC <sup>1</sup>
\$0005	PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0	PORTD <sup>2</sup>
\$0006	Bit 7	6	5	4	3	2	1	Bit 0	DDRC <sup>1</sup>
\$0007	Bit 7	6	5	4	3	2	1	Bit 0	DDRD <sup>2</sup>
\$0008	PE7	PE6	PE5	PE4	PE3	PE2	PE1	PE0	PORTE <sup>3</sup>
\$0009	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	0	0	DDRE <sup>3</sup>
\$000A	ARSIE	PLLTE	PIPOE	NECLK	LSTRE	RDWE	0	0	PEAR <sup>4</sup>
\$000B	SMODN	MODB	MODA	ESTR	IVIS	0	EMD	EME	MODE <sup>4</sup>
\$000C	PUPH	PUPG	PUPF	PUPE	PUPD	PUPC	PUPB	PUPA	PUCR <sup>4</sup>
\$000D	RDPJ	RDPH	RDPG	RDPF	RDPE	RDPD	RDPC	RDPAB	RDRIV <sup>4</sup>
\$000E	0	0	0	0	0	0	0	0	Reserved <sup>4</sup>
\$000F	0	0	0	0	0	0	0	0	Reserved <sup>4</sup>
\$0010	RAM15	RAM14	RAM13	RAM12	RAM11	0	0	0	INITRM
\$0011	REG15	REG14	REG13	REG12	REG11	0	0	0	INITRG
\$0012	EE15	EE14	EE13	EE12	0	0	0	EEON	INITEE
\$0013	EWDIR	NDRC	0	0	0	0	0	0	MISC
\$0014	RTIE	RSWAI	RSBCK	0	RTBYP	RTR2	RTR1	RTR0	RTICTL
\$0015	RTIF	0	0	0	0	0	0	0	RTIFLG
\$0016	CME	FCME	FCM	FCOP	DISR	CR2	CR1	CR0	COPCTL
\$0017	Bit 7	6	5	4	3	2	1	Bit 0	COPRST
\$0018	ITE6	ITE8	ITEA	ITEC	ITEE	ITF0	ITF2	ITF4	ITST0
\$0019	ITD6	ITD8	ITDA	ITDC	ITDE	ITE0	ITE2	ITE4	ITST1
\$001A	ITC6	ITC8	ITCA	ITCC	ITCE	ITD0	ITD2	ITD4	ITST2
\$001B	0	0	0	0	0	ITC0	ITC2	ITC4	ITST3
\$001C	0	0	0	0	0	0	0	0	Reserved
\$001D	0	0	0	0	0	0	0	0	Reserved
\$001E	IRQE	IRQEN	DLY	0	0	0	0	0	INTCR
\$001F	1	1	PSEL5	PSEL4	PSEL3	PSEL2	PSEL1	0	HPRIO
\$0020	Bit 7	6	5	4	3	2	1	Bit 0	KWIED <sup>5</sup>
\$0021	Bit 7	6	5	4	3	2	1	Bit 0	KWIFD <sup>5</sup>
\$0022	0	0	0	0	0	0	0	0	Reserved
\$0023	0	0	0	0	0	0	0	0	Reserved
\$0024	PH7	PH6	PH5	PH4	PH3	PH2	PH1	PH0	PORTH
\$0025	Bit 7	6	5	4	3	2	1	Bit 0	DDRH
\$0026	Bit 7	6	5	4	3	2	1	Bit 0	KWIEH
\$0027	Bit 7	6	5	4	3	2	1	Bit 0	KWIFH
\$0028	PJ7	PJ6	PJ5	PJ4	PJ3	PJ2	PJ1	PJ0	PORTJ

**Table 7 MC68HC812A4 Register Map (Sheet 2 of 4)**

Address	Bit 7	6	5	4	3	2	1	Bit 0	Name
\$0029	Bit 7	6	5	4	3	2	1	Bit 0	DDRJ
\$002A	Bit 7	6	5	4	3	2	1	Bit 0	KWIEJ
\$002B	Bit 7	6	5	4	3	2	1	Bit 0	KWIFJ
\$002C	Bit 7	6	5	4	3	2	1	Bit 0	KPOLJ
\$002D	Bit 7	6	5	4	3	2	1	Bit 0	PUPSJ
\$002E	Bit 7	6	5	4	3	2	1	Bit 0	PULEJ
\$002F	0	0	0	0	0	0	0	0	Reserved
\$0030	0	PF6	PF5	PF4	PF3	PF2	PF1	PF0	PORTF
\$0031	0	0	PG5	PG4	PG3	PG2	PG1	PG0	PORTG
\$0032	0	Bit 6	5	4	3	2	1	Bit 0	DDRF
\$0033	0	0	Bit 5	4	3	2	1	Bit 0	DDRG
\$0034	PDA19	PDA18	PDA17	PDA16	PDA15	PDA14	PDA13	PDA12	DPAGE
\$0035	PPA21	PPA20	PPA19	PPA18	PPA17	PPA16	PPA15	PPA14	PPAGE
\$0036	PEA17	PEA16	PEA15	PEA14	PEA13	PEA12	PEA11	PEA10	EPAGE
\$0037	DWEN	PWEN	EWEN	0	0	0	0	0	WINDEF
\$0038	0	0	A21E	A20E	A19E	A18E	A17E	A16E	MXAR
\$0039	0	0	0	0	0	0	0	0	Reserved
\$003A	0	0	0	0	0	0	0	0	Reserved
\$003B	0	0	0	0	0	0	0	0	Reserved
\$003C	0	CSP1E	CSP0E	CSDE	CS3E	CS2E	CS1E	CS0E	CSCTL0
\$003D	0	CSP1FL	CSPA21	CSDHF	CS3EP	0	0	0	CSCTL1
\$003E	0	0	SRP1A	SRP1B	SRP0A	SRP0B	STRDA	STRDB	CSSTR0
\$003F	STR3A	STR3B	STR2A	STR2B	STR1A	STR1B	STR0A	STR0B	CSSTR1
\$0040	0	0	0	0	LDV11	LDV10	LDV9	LDV8	LDV
\$0041	LDV7	LDV6	LDV5	LDV4	LDV3	LDV2	LDV1	LDV0	LDV
\$0042	0	0	0	0	RDV11	RDV10	RDV9	RDV8	RDV
\$0043	RDV7	RDV6	RDV5	RDV4	RDV3	RDV2	RDV1	RDV0	RDV
\$0044	0	0	0	0	0	0	0	0	Reserved
\$0045	0	0	0	0	0	0	0	0	Reserved
\$0046	0	0	0	0	0	0	0	0	Reserved
\$0047	LCK	PLLON	PLLS	BCSC	BCSB	BCSA	MCSB	MCSA	CLKCTL
\$0048– \$005F	0	0	0	0	0	0	0	0	Reserved
\$0060	0	0	0	0	0	0	0	0	ATDCTL0
\$0061	0	0	0	0	0	0	0	0	ATDCTL1
\$0062	ADPU	AFFC	AWAI	0	0	0	ASCIE	ASCIF	ATDCTL2
\$0063	0	0	0	0	0	0	FRZ1	FRZ0	ATDCTL3
\$0064	0	SMP1	SMP0	PRS4	PRS3	PRS2	PRS1	PRS0	ATDCTL4
\$0065	0	S8CM	SCAN	MULT	CD	CC	CB	CA	ATDCTL5
\$0066	SCF	0	0	0	0	CC2	CC1	CC0	ATDSTAT
\$0067	CCF7	CCF6	CCF5	CCF4	CCF3	CCF2	CCF1	CCF0	ATDSTAT
\$0068	SAR9	SAR8	SAR7	SAR6	SAR5	SAR4	SAR3	SAR2	ATDTEST
\$0069	SAR1	SAR0	RST	TSTOUT	TST3	TST2	TST1	TST0	ATDTEST
\$006A– \$006E	0	0	0	0	0	0	0	0	Reserved
\$006F	PAD7	PAD6	PAD5	PAD4	PAD3	PAD2	PAD1	PAD0	PORTAD
\$0070	Bit 7	6	5	4	3	2	1	Bit 0	ADR0H
\$0071	0	0	0	0	0	0	0	0	Reserved
\$0072	Bit 7	6	5	4	3	2	1	Bit 0	ADR1H

**Table 7 MC68HC812A4 Register Map (Sheet 3 of 4)**

Address	Bit 7	6	5	4	3	2	1	Bit 0	Name
\$0073	0	0	0	0	0	0	0	0	Reserved
\$0074	Bit 7	6	5	4	3	2	1	Bit 0	ADR2H
\$0075	0	0	0	0	0	0	0	0	Reserved
\$0076	Bit 7	6	5	4	3	2	1	Bit 0	ADR3H
\$0077	0	0	0	0	0	0	0	0	Reserved
\$0078	Bit 7	6	5	4	3	2	1	Bit 0	ADR4H
\$0079	0	0	0	0	0	0	0	0	Reserved
\$007A	Bit 7	6	5	4	3	2	1	Bit 0	ADR5H
\$007B	0	0	0	0	0	0	0	0	Reserved
\$007C	Bit 7	6	5	4	3	2	1	Bit 0	ADR6H
\$007D	0	0	0	0	0	0	0	0	Reserved
\$007E	Bit 7	6	5	4	3	2	1	Bit 0	ADR7H
\$007F	0	0	0	0	0	0	0	0	Reserved
\$0080	IOS7	IOS6	IOS5	IOS4	IOS3	IOS2	IOS1	IOS0	TIOS
\$0081	FOC7	FOC6	FOC5	FOC4	FOC3	FOC2	FOC1	FOC0	CFORC
\$0082	OC7M7	OC7M6	OC7M5	OC7M4	OC7M3	OC7M2	OC7M1	OC7M0	OC7M
\$0083	OC7D7	OC7D6	OC7D5	OC7D4	OC7D3	OC7D2	OC7D1	OC7D0	OC7D
\$0084	Bit 15	14	13	12	11	10	9	Bit 8	TCNT
\$0085	Bit 7	6	5	4	3	2	1	Bit 0	TCNT
\$0086	TEN	TSWAI	TSBCK	TFFCA	PAOQE	T7QE	T1QE	T0QE	TSCR
\$0087	PAOQB	PAOQA	T7QB	T7QA	T1QB	T1QA	T0QB	T0QA	TQCR
\$0088	OM7	OL7	OM6	OL6	OM5	OL5	OM4	OL4	TCTL1
\$0089	OM3	OL3	OM2	OL2	OM1	OL1	OM0	OL0	TCTL2
\$008A	EDG7B	EDG7A	EDG6B	EDG6A	EDG5B	EDG5A	EDG4B	EDG4A	TCTL3
\$008B	EDG3B	EDG3A	EDG2B	EDG2A	EDG1B	EDG1A	EDG0B	EDG0A	TCTL4
\$008C	C7I	C6I	C5I	C4I	C3I	C2I	C1I	C0I	TMSK1
\$008D	TOI	0	TPU	TDRB	TCRE	PR2	PR1	PR0	TMSK2
\$008E	C7F	C6F	C5F	C4F	C3F	C2F	C1F	C0F	TFLG1
\$008F	TOF	0	0	0	0	0	0	0	TFLG2
\$0090	Bit 15	14	13	12	11	10	9	Bit 8	TC0
\$0091	Bit 7	6	5	4	3	2	1	Bit 0	TC0
\$0092	Bit 15	14	13	12	11	10	9	Bit 8	TC1
\$0093	Bit 7	6	5	4	3	2	1	Bit 0	TC1
\$0094	Bit 15	14	13	12	11	10	9	Bit 8	TC2
\$0095	Bit 7	6	5	4	3	2	1	Bit 0	TC2
\$0096	Bit 15	14	13	12	11	10	9	Bit 8	TC3
\$0097	Bit 7	6	5	4	3	2	1	Bit 0	TC3
\$0098	Bit 15	14	13	12	11	10	9	Bit 8	TC4
\$0099	Bit 7	6	5	4	3	2	1	Bit 0	TC4
\$009A	Bit 15	14	13	12	11	10	9	Bit 8	TC5
\$009B	Bit 7	6	5	4	3	2	1	Bit 0	TC5
\$009C	Bit 15	14	13	12	11	10	9	Bit 8	TC6
\$009D	Bit 7	6	5	4	3	2	1	Bit 0	TC6
\$009E	Bit 15	14	13	12	11	10	9	Bit 8	TC7
\$009F	Bit 7	6	5	4	3	2	1	Bit 0	TC7
\$00A0	0	PAEN	PAMOD	PEDGE	CLK1	CLK0	PAOVI	PAI	PACTL
\$00A1	0	0	0	0	0	0	PAOVF	PAIF	PAFLG
\$00A2	Bit 15	14	13	12	11	10	9	Bit 8	PACNT
\$00A3	Bit 7	6	5	4	3	2	1	Bit 0	PACNT

**Table 7 MC68HC812A4 Register Map (Sheet 4 of 4)**

Address	Bit 7	6	5	4	3	2	1	Bit 0	Name
\$00A4– \$00AC	0	0	0	0	0	0	0	0	Reserved
\$00AD	0	0	0	0	0	0	TCBYP	PCBYP	TIMTST
\$00AE	PT7	PT6	PT5	PT4	PT3	PT2	PT1	PT0	PORTT
\$00AF	Bit 7	6	5	4	3	2	1	Bit 0	DDRT
\$00B0– \$00BF	0	0	0	0	0	0	0	0	Reserved
\$00C0	BTST	BSPL	BRLD	SBR12	SBR11	SBR10	SBR9	SBR8	SC0BDH
\$00C1	SBR7	SBR6	SBR5	SBR4	SBR3	SBR2	SBR1	SBR0	SC0BDL
\$00C2	LOOPS	WOMS	RSRC	M	WAKE	ILT	PE	PT	SC0CR1
\$00C3	TIE	TCIE	RIE	ILIE	TE	RE	RWU	SBK	SC0CR2
\$00C4	TDRE	TC	RDRF	IDLE	OR	NF	FE	PF	SC0SR1
\$00C5	0	0	0	0	0	0	0	RAF	SC0SR2
\$00C6	R8	T8	0	0	0	0	0	0	SC0DRH
\$00C7	R7T7	R6T6	R5T5	R4T4	R3T3	R2T2	R1T1	R0T0	SC0DRL
\$00C8	BTST	BSPL	BRLD	SBR12	SBR11	SBR10	SBR9	SBR8	SC1BDH
\$00C9	SBR7	SBR6	SBR5	SBR4	SBR3	SBR2	SBR1	SBR0	SC1BDL
\$00CA	LOOPS	WOMS	RSRC	M	WAKE	ILT	PE	PT	SC1CR1
\$00CB	TIE	TCIE	RIE	ILIE	TE	RE	RWU	SBK	SC1CR2
\$00CC	TDRE	TC	RDRF	IDLE	OR	NF	FE	PF	SC1SR1
\$00CD	0	0	0	0	0	0	0	RAF	SC1SR2
\$00CE	R8	T8	0	0	0	0	0	0	SC1DRH
\$00CF	R7T7	R6T6	R5T5	R4T4	R3T3	R2T2	R1T1	R0T0	SC1DRL
\$00D0	SPIE	SPE	SWOM	MSTR	CPOL	CPHA	SSOE	LSBF	SP0CR1
\$00D1	SPFQE	SPFQB	SPFQA	0	PUPS	RDS	0	SPC0	SP0CR2
\$00D2	0	0	0	0	0	SPR2	SPR1	SPR0	SP0BR
\$00D3	SPIF	WCOL	0	MODF	0	0	0	0	SP0SR
\$00D4	0	0	0	0	0	0	0	0	Reserved
\$00D5	Bit 7	6	5	4	3	2	1	Bit 0	SP0DR
\$00D6	PS7	PS6	PS5	PS4	PS3	PS2	PS1	PS0	PORTS
\$00D7	Bit 7	6	5	4	3	2	1	Bit 0	DDRS
\$00D8– \$00DF	0	0	0	0	0	0	0	0	Reserved
\$00E0– \$00EF	0	0	0	0	0	0	0	0	Reserved
\$00F0	1	1	1	1	1	1	PROTLCK	EERC	EEMCR
\$00F1	1	BPROT6	BPROT5	BPROT4	BPROT3	BPROT2	BPROT1	BPROT0	EEPROT
\$00F2	EEODD	EEVEN	MARG	EECPD	EECPRD	0	EECPM	0	EETST
\$00F3	BULKP	0	0	BYTE	ROW	ERASE	EELAT	EEPGM	EEPROM
\$00F4– \$01FF	0	0	0	0	0	0	0	0	Reserved

1. Port A, port B, port C and data direction registers DDRA, DDRB, and DDRC are not in map in expanded and peripheral modes.
2. Port D and DDRD not in map in wide expanded modes and peripheral mode; also not in map in narrow special expanded mode with EMD set.
3. Port E and DDRE not in map in peripheral mode; also not in map in expanded modes with EME set.
4. Registers also not in map in peripheral mode.
5. Key wake-up associated with port D not in map in wide expanded modes; also not in map in narrow special expanded mode with EMD set.

## 5 Bus Control and Input/Output

Internally the MC68HC812A4 has full 16-bit data paths, but depending upon the operating mode and control registers, the external bus may be 8 or 16 bits. There are cases where 8-bit and 16-bit accesses can appear on adjacent cycles using the  $\overline{\text{LSTRB}}$  signal to indicate 8- or 16-bit data.

### 5.1 Detecting Access Type from External Signals

The external signals  $\overline{\text{LSTRB}}$ ,  $\overline{\text{R/W}}$ , and  $\text{A0}$  can be used to determine the type of bus access that is taking place. Accesses to the internal RAM module are the only type of access that produce  $\overline{\text{LSTRB}} = \text{A0} = 1$ , because the internal RAM is specifically designed to allow misaligned 16-bit accesses in a single cycle. In these cases the data for the address that was accessed is on the low half of the data bus and the data for address + 1 is on the high half of the data bus.

**Table 8 Access Type vs. Bus Control Pins**

$\overline{\text{LSTRB}}$	$\text{A0}$	$\overline{\text{R/W}}$	Type of Access
1	0	1	8-bit read of an even address
0	1	1	8-bit read of an odd address
1	0	0	8-bit write of an even address
0	1	0	8-bit write of an odd address
0	0	1	16-bit read of an even address
1	1	1	16-bit read of an odd address (low/high data swapped)
0	0	0	16-bit write to an even address
1	1	0	16-bit write to an even address (low/high data swapped)

### 5.2 Registers

Not all registers are visible in the MC68HC812A4 memory map under certain conditions. In special peripheral mode the first 16 registers associated with bus expansion are removed from the memory map.

In expanded modes, some or all of port A, port B, port C, port D, and port E are used for expansion buses and control signals. In order to allow emulation of the single-chip functions of these ports, some of these registers must be rebuilt in an external port replacement unit. In any expanded mode, port A, port B, and port C are used for address and data lines so registers for these ports, as well as the data direction registers for these ports, are removed from the on-chip memory map and become external accesses.

Port D and its associated data direction register may be removed from the on-chip map when port D is needed for 16-bit data transfers. If the MCU is in an expanded wide mode, port C and port D are used for 16-bit data and the associated port and data direction registers become external accesses. When the MCU is in expanded narrow mode, the external data bus is normally 8-bits. To allow full-speed operation while allowing visibility of internal 16-bit accesses, a 16-bit-wide data path is required. The emulate port D (EMD) control bit in the MODE register may be set to allow such 16-bit transfers. In this case of narrow special expanded mode and the EMD bit set, port D and data direction D registers are removed from the on-chip memory map and become external accesses so port D may be rebuilt externally.

In any expanded mode, port E pins may be needed for bus control (e.g.,  $\text{ECLK}$ ,  $\overline{\text{R/W}}$ ). To regain the single-chip functions of port E, the emulate port E (EME) control bit in the MODE register may be set. In this special case of expanded mode and EME set, PORTE and DDRE registers are removed from the on-chip memory map and become external accesses so port E may be rebuilt externally.

**PORTA** — Port A Register

**\$0000**

	Bit 7	6	5	4	3	2	1	Bit 0
Single Chip	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0
RESET:	0	0	0	0	0	0	0	0
Expanded & Periph:	ADDR15	ADDR14	ADDR13	ADDR12	ADDR11	ADDR10	ADDR9	ADDR8

Bits PA[7:0] are associated with addresses ADDR[15:8] respectively. When this port is not used for external addresses such as in single-chip mode, these pins can be used as general-purpose I/O. DDRA determines the primary direction of each pin. This register is not in the on-chip map in expanded and peripheral modes. Read and write anytime.

**DDRA** — Port A Data Direction Register

**\$0002**

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

This register determines the primary direction for each port A pin when functioning as a general-purpose I/O port. DDRA is not in the on-chip map in expanded and peripheral modes. Read and write anytime.  
0 = Associated pin is a high-impedance input  
1 = Associated pin is an output

**PORTB** — Port B Register

**\$0001**

	Bit 7	6	5	4	3	2	1	Bit 0
Single Chip	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0
RESET:	0	0	0	0	0	0	0	0
Expanded & Periph:	ADDR7	ADDR6	ADDR5	ADDR4	ADDR3	ADDR2	ADDR1	ADDR0

Bits PB[7:0] are associated with addresses ADDR[7:0] respectively. When this port is not used for external addresses such as in single-chip mode, these pins can be used as general-purpose I/O. DDRB determines the primary direction of each pin. This register is not in the on-chip map in expanded and peripheral modes. Read and write anytime.

**DDRB** — Port B Data Direction Register

**\$0003**

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

This register determines the primary direction for each port B pin when functioning as a general-purpose I/O port. DDRB is not in the on-chip map in expanded and peripheral modes. Read and write anytime.  
0 = Associated pin is a high-impedance input  
1 = Associated pin is an output

**PORTC** — Port C Register**\$0004**

	Bit 7	6	5	4	3	2	1	Bit 0
Single Chip	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0
RESET:	0	0	0	0	0	0	0	0
Exp Wide & Periph	DATA15	DATA14	DATA13	DATA12	DATA11	DATA10	DATA9	DATA8
Expanded Narrow	DATA15/7	DATA14/6	DATA13/5	DATA12/4	DATA11/3	DATA10/2	DATA9/1	DATA8/0

Bits PC[7:0] are associated with DATA[15:8] respectively. When this port is not used for external data such as in single-chip mode, these pins can be used as general-purpose I/O. DDRC determines the primary direction for each pin. In narrow expanded modes, DATA[15:8] and DATA[7:0] are multiplexed into the MCU through port C pins on successive cycles. This register is not in the on-chip map in expanded and peripheral modes.

Read and write anytime (provided this register is in the map).

When the MCU is operating in special expanded narrow mode and port C and port D are being used for internal visibility, internal accesses produce full 16-bit information with DATA[15:8] on port C and DATA[7:0] on port D. This allows the MCU to operate at full speed while making 16-bit access information available to external development equipment in a single cycle. In this narrow mode, normal 16-bit accesses to external memory get split into two successive 8-bit accesses on port C alone.

**DDRC** — Port C Data Direction Register**\$0006**

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

This register determines the primary direction for each port C pin when functioning as a general-purpose I/O port. DDRC is not in the on-chip map in expanded and peripheral modes. Read and write anytime.

0 = Associated pin is a high-impedance input

1 = Associated pin is an output

**PORTD** — Port D Register**\$0005**

	Bit 7	6	5	4	3	2	1	Bit 0
Sgl Chip & Exp Narrow	PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0
RESET:	0	0	0	0	0	0	0	0
Exp Wide & Periph	DATA7	DATA6	DATA5	DATA4	DATA3	DATA2	DATA1	DATA0
Alt. Pin Function	KWD7	KWD6	KWD5	KWD4	KWD3	KWD2	KWD1	KWD0

Bits PD[7:0] are associated with DATA[7:0] respectively. When this port is not used for external data, such as in single-chip mode, these pins can be used as general-purpose I/O or key wakeup signals. DDRD determines the primary direction of each port D pin.

In special expanded narrow mode the external data bus is normally limited to eight bits on port C but the emulate port D (EMD) control bit in the MODE register can be set to allow port C and port D to be used together to provide single-cycle visibility of internal 16-bit accesses for debugging purposes. If the mode is special narrow expanded and EMD is set, port D is configured for DATA[7:0] of visible internal accesses and normal 16-bit external accesses are split into two adjacent 8-bit accesses through port C. This allows connection of a single 8-bit external program memory.

This register is not in the on-chip map in wide expanded and peripheral modes. Also, in special narrow expanded mode, the function of this port is determined by the EMD control bit. If EMD is set, this register is not in the on-chip map and port D is used for DATA[7:0] of visible internal accesses. If EMD is clear, this port serves as general-purpose I/O or key wakeup signals. Read and write anytime.

**DDRD** — Port D Data Direction Register

**\$0007**

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

When port D is operating as a general-purpose I/O port, this register determines the primary direction for each port D pin.

0 = Associated pin is a high-impedance input

1 = Associated pin is an output

This register is not in the map in wide expanded and peripheral modes. Also, in special narrow expanded mode, the function of this port is determined by the EMD control bit. If EMD is set, this register is not in the on-chip map and port D is used for DATA[7:0] of visible internal accesses. If EMD is clear, this port serves as general-purpose I/O or key wakeup signals.

Read and write anytime.

**PORTE** — Port E Register

**\$0008**

	Bit 7	6	5	4	3	2	1	Bit 0	
Single Chip	PE7	PE6	PE5	PE4	PE3	PE2	PE1	PE0	
RESET:	0	0	0	0	1	0	0	0	Normal Narrow Expanded
RESET:	0	0	0	0	0	0	0	0	All other modes
Alt. Pin Function	ARST	MODB or IPIPE1	MODA or IPIPE0	ECLK	$\overline{\text{LSTRB}}$	R/ $\overline{\text{W}}$	$\overline{\text{IRQ}}$	$\overline{\text{XIRQ}}$	

This register is associated with external bus control signals and interrupt inputs including auxiliary reset (ARST), mode select (MODB/IPIPE1, MODA/IPIPE0), E clock, size ( $\overline{\text{LSTRB}}$ ), read/write (R/ $\overline{\text{W}}$ ),  $\overline{\text{IRQ}}$ , and  $\overline{\text{XIRQ}}$ . When the associated pin is not used for one of these specific functions, the pin can be used as general-purpose I/O. The port E assignment register (PEAR) selects the function of each pin. DDRD determines the primary direction of each port E pin when configured to be general-purpose I/O.

Some of these pins have software selectable pull-ups ( $\overline{\text{LSTRB}}$ , R/ $\overline{\text{W}}$ , and  $\overline{\text{XIRQ}}$ ). A single control bit enables the pull-ups for all these pins which are configured as inputs.  $\overline{\text{IRQ}}$  always has a pull-up.

PE7 can be selected as a high-true auxiliary reset input.

This register is not in the map in peripheral mode or expanded modes when the EME bit is set.

Read and write anytime.

**DDRE — Port E Data Direction Register**
**\$0009**

	Bit 7	6	5	4	3	2	1	Bit 0	
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	0	0	
RESET:	0	0	0	0	1	0	0	0	Normal Narrow Expanded
RESET:	0	0	0	0	0	0	0	0	All other Modes

This register determines the primary direction for each port E pin configured as general-purpose I/O.

0 = Associated pin is a high-impedance input

1 = Associated pin is an output

PE[1:0] are associated with  $\overline{XIRQ}$  and  $\overline{IRQ}$  and cannot be configured as outputs. These pins can be read regardless of whether the alternate interrupt functions are enabled.

This register is not in the map in peripheral mode and expanded modes while the EME control bit is set.

Read and write anytime.

**PEAR — Port E Assignment Register**
**\$000A**

	Bit 7	6	5	4	3	2	1	Bit 0	
	ARSIE	PLLTE	PIPOE	NECLK	LSTRE	RDWE	0	0	
RESET:	0	0	1	0	1	1	0	0	Special Single Chip
RESET:	0	0	1	0	1	1	0	0	Special Exp Nar
RESET:	0	1	0	1	0	0	0	0	Peripheral
RESET:	0	0	1	0	1	1	0	0	Special Exp Wide
RESET:	0	0	0	1	0	0	0	0	Normal Single Chip
RESET:	0	0	0	0	0	0	0	0	Normal Exp Nar
RESET:	0	0	0	0	0	0	0	0	Normal Exp Wide

The PEAR register is used to choose between the general-purpose I/O functions and the alternate bus control functions of port E. When an alternate control function is selected, the associated DDRE bits are overridden.

The reset condition of this register depends on the mode of operation because bus-control signals are needed immediately after reset in some modes. In normal single-chip mode, no external bus control signals are needed so all of port E is configured for general-purpose I/O. In special single-chip mode, the E clock is enabled as a timing reference and the other bits of port E are configured for general-purpose I/O. In normal expanded modes, the reset vector is located in external memory. The E clock may be required for this access but  $R/\overline{W}$  is only needed by the system when there are external writable resources. Therefore in normal expanded modes, only the E clock is configured for its alternate bus control function and the other bits of port E are configured for general-purpose I/O. If the normal expanded system needs any other bus-control signals, PEAR would need to be written before any access that needed the additional signals. In special expanded modes, IPIPE1, IPIPE0, E,  $R/\overline{W}$ , and  $\overline{LSTRB}$  are configured as bus-control signals.

In peripheral mode, the PEAR register is not accessible for reads or writes. However, the PLLTE control bit is reset to one to configure PE6 as a test output from the PLL module.

**ARSIE** — Auxiliary Reset Input Enable

Read and write anytime.

0 = PE7 is general-purpose I/O.

1 = PE7 is a high-true reset input. Reset timing is the same as that of the low-true  $\overline{\text{RESET}}$  pin.

**PLLTE** — PLL Testing Enable

Normal: write never; Special: write anytime EXCEPT the first time. Read anytime.

0 = PE6 is general-purpose I/O or pipe output.

1 = PE6 is a test signal output from the PLL module (no effect in single chip or normal expanded modes).  $\text{PIPOE} = 1$  overrides this function and forces PE6 to be a pipe status output signal.

**PIPOE** — Pipe Status Signal Output Enable

Normal: write once; Special: write anytime except the first time. Read anytime.

0 = PE[6:5] are general-purpose I/O (if  $\text{PLLTE} = 1$ , PE6 is a test output signal from the PLL module).

1 = PE[6:5] are outputs and indicate the state of the instruction queue (no effect in single chip modes).

**NECLK** — No External E Clock

Normal: write anytime; Special: write never. Read anytime. In peripheral mode, E is an input and in all other modes, E is an output.

0 = PE4 is the external E-clock pin. To get a free-running E clock in single-chip modes, use  $\text{NECLK}=0$  and  $\text{IVIS}=1$ . A 16-bit write to PEAR:MODE can configure these bits in one operation.

1 = PE4 is a general-purpose I/O pin.

**LSTRE** — Low Strobe ( $\overline{\text{LSTRB}}$ ) Enable

Normal: write once; Special: write anytime except the first time. Read anytime. This bit has no effect in single-chip modes or normal expanded narrow mode.

0 = PE3 is a general-purpose I/O pin.

1 = PE3 is configured as the  $\overline{\text{LSTRB}}$  bus-control output, provided the MCU is not in single chip or normal expanded narrow modes.

$\overline{\text{LSTRB}}$  is used during external writes. After reset in normal expanded mode,  $\overline{\text{LSTRB}}$  is disabled. If needed, it should be enabled before external writes. External reads do not normally need  $\overline{\text{LSTRB}}$  because all 16 data bits can be driven even if the MCU only needs eight bits of data.

In normal expanded narrow mode this pin is reset to an output driving high allowing the pin to be an output while in and immediately after reset.

**RDWE** — Read/Write Enable

Normal: write once; Special: write anytime except the first time. Read anytime. This bit has no effect in single-chip modes.

0 = PE2 is a general-purpose I/O pin.

1 = PE2 is configured as the  $\text{R}/\overline{\text{W}}$  pin. In single chip modes, RDWE has no effect and PE2 is a general-purpose I/O pin.

$\text{R}/\overline{\text{W}}$  is used for external writes. After reset in normal expanded mode, it is disabled. If needed it should be enabled before any external writes.

**PUCR** — Pull Up Control Register**\$000C**

	Bit 7	6	5	4	3	2	1	Bit 0
	PUPH	PUPG	PUPF	PUPE	PUPD	PUPC	PUPB	PUPA
RESET:	1	1	1	1	1	1	1	1

These bits select pull-up resistors for any pin in the corresponding port that is currently configured as an input. This register is not in the map in peripheral mode.

Read and write anytime.

**PUPH** — Pull-Up Port H Enable

- 0 = Port H pull-ups are disabled.
- 1 = Enable pull-up devices for all port H input pins.

**PUPG** — Pull-Up Port G Enable

- 0 = Port G pull-ups are disabled.
- 1 = Enable pull-up devices for all port G input pins.

**PUPF** — Pull-Up Port F Enable

- 0 = Port F pull-ups are disabled.
- 1 = Enable pull-up devices for all port F input pins.

**PUPE** — Pull-Up Port E Enable

- 0 = Port E pull-ups on PE3, PE2, and PE0 are disabled.
- 1 = Enable pull-up devices for port E input pins PE3, PE2, and PE0.

**PUPD** — Pull-Up Port D Enable

- 0 = Port D pull-ups are disabled.
- 1 = Enable pull-up devices for all port D input pins.

This bit has no effect if port D is being used as part of the data bus (the pull-ups are inactive).

**PUPC** — Pull-Up Port C Enable

- 0 = Port C pull-ups are disabled.
- 1 = Enable pull-up devices for all port C input pins.

This bit has no effect if port C is being used as part of the data bus (the pull-ups are inactive).

**PUPB** — Pull-Up Port B Enable

- 0 = Port B pull-ups are disabled.
- 1 = Enable pull-up devices for all port B input pins.

This bit has no effect if port B is being used as part of the address bus (the pull-ups are inactive).

**PUPA** — Pull-Up Port A Enable

- 0 = Port A pull-ups are disabled.
- 1 = Enable pull-up devices for all port A input pins.

This bit has no effect if port A is being used as part of the address bus (the pull-ups are inactive).

**RDRIV** — Reduced Drive of I/O Lines**\$000D**

	Bit 7	6	5	4	3	2	1	Bit 0
	RDPJ	RDPH	RDPG	RDPF	RDPE	RDPD	RDPC	RDPAB
RESET:	0	0	0	0	0	0	0	0

These bits select reduced drive for the associated port pins. This gives reduced power consumption and reduced RFI with a slight increase in transition time (depending on loading). The reduced drive function is independent of which function is being used on a particular port.

This register is not in the map in peripheral mode.

Normal: write anytime; Special: write never. Read anytime.

**RDPJ** — Reduced Drive of Port J

- 0 = All port J output pins have full drive enabled.
- 1 = All port J output pins have reduced drive capability.

**RDPH** — Reduced Drive of Port H

- 0 = All port H output pins have full drive enabled.
- 1 = All port H output pins have reduced drive capability.

**RDPG** — Reduced Drive of Port G

- 0 = All port G output pins have full drive enabled.
- 1 = All port G output pins have reduced drive capability.

**RDPF** — Reduced Drive of Port F

- 0 = All port F output pins have full drive enabled.
- 1 = All port F output pins have reduced drive capability.

**RDPE** — Reduced Drive of Port E

- 0 = All port E output pins have full drive enabled.
- 1 = All port E output pins have reduced drive capability.

**RDPD** — Reduced Drive of Port D

- 0 = All port D output pins have full drive enabled.
- 1 = All port D output pins have reduced drive capability.

**RDPC** — Reduced Drive of Port C

- 0 = All port C output pins have full drive enabled.
- 1 = All port C output pins have reduced drive capability.

**RDPAB** — Reduced Drive of Port A and Port B

- 0 = All port A and port B output pins have full drive enabled.
- 1 = All port A and port B output pins have reduced drive capability.

## 6 Operating Modes and Resource Mapping

Eight possible operating modes determine the operating configuration of the MC68HC812A4. Each mode has an associated default memory map and external bus configuration. After reset, most system resources can be mapped to other addresses by writing to the appropriate control registers.

### 6.1 Operating Modes

The operating mode out of reset is determined by the states of the BKGD, MODB, and MODA pins during reset.

The SMODN, MODB, and MODA bits in the MODE register show current operating mode and provide limited mode switching during operation. The states of the BKGD, MODB, and MODA pins are latched into these bits on the rising edge of the reset signal.

**Table 9 Mode Selection**

BKGD	MODB	MODA	Mode	Port A Port B	Port C	Port D
0	0	0	Special Single Chip	G.P. I/O	G.P. I/O	G.P. I/O
0	0	1	Special Expanded Narrow	ADDR	DATA	G.P. I/O
0	1	0	Special Peripheral	ADDR	DATA	DATA
0	1	1	Special Expanded Wide	ADDR	DATA	DATA
1	0	0	Normal Single Chip	G.P. I/O	G.P. I/O	G.P. I/O
1	0	1	Normal Expanded Narrow	ADDR	DATA	G.P. I/O
1	1	0	Reserved (Forced to Peripheral)	—	—	—
1	1	1	Normal Expanded Wide	ADDR	DATA	DATA

There are two basic types of operating modes:

**Normal** modes — some registers and bits are protected against accidental changes.

**Special** modes — allow greater access to protected control registers and bits for special purposes such as testing and emulation.

A system development and debug feature, background debug mode (BDM), is available in all modes. In special single-chip mode, BDM is active immediately after reset.

#### 6.1.1 Normal Operating Modes

These modes provide three operating configurations. Background debugging is available in all three modes, but must first be enabled for some operations by means of a BDM command. BDM can then be made active by another BDM command.

**Normal Expanded Wide Mode** — This is a normal mode of operation in which the expanded bus is present with a 16-bit data bus. Ports A and B are used for the 16-bit address bus. Ports C and D are used for the 16-bit data bus.

**Normal Expanded Narrow Mode** — This is a normal mode of operation in which the expanded bus is present with an 8-bit data bus. Ports A and B are used for the 16-bit address bus. Port C is used as the data bus. In this mode, 16-bit data is presented one byte at a time, the high byte followed by the low byte. The address is automatically incremented on the second cycle.

**Normal Single-Chip Mode** — There are no external address and data buses in this mode. The MCU operates as a stand-alone device and all program and data resources are on-chip. External port pins normally associated with address and data buses can be used for general-purpose I/O.

### 6.1.2 Special Operating Modes

There are three special operating modes that correspond to normal operating modes. These operating modes are commonly used in factory testing and system development. In addition, there is a special peripheral mode, in which an external master, such as an I.C. tester, can control the on-chip peripherals.

**Special Expanded Wide Mode** — This mode can be used for emulation of normal expanded wide mode and emulation of normal single-chip mode. Port A and port B are used for a 16-bit address bus. Port C and port D are used for a 16-bit data bus.

**Special Expanded Narrow Mode** — This mode can be used for emulation of normal expanded narrow mode. Port A and port B are used for the 16-bit address bus. Port C is used as the data bus. In this mode external 16-bit data is handled as two back-to-back bus cycles, one for the high byte followed by one for the low byte. For development purposes, port D can be made available for visibility of 16-bit internal accesses by setting the EMD and IVIS control bits.

**Special Single-Chip Mode** — This mode can be used to force the MCU to active BDM mode to allow system debug through the BKGD pin. There are no external address and data buses in this mode. The MCU operates as a stand-alone device and all program and data space are on-chip. External port pins can be used for general-purpose I/O.

**Special Peripheral Mode** — The CPU is not active in this mode. An external master can control on-chip peripherals for testing purposes. It is not possible to change to or from this mode without going through reset. Background debugging should not be used while the MCU is in special peripheral mode as internal bus conflicts between BDM and the external master can cause improper operation of both modes.

## 6.2 Background Debug Mode

Background debug mode (BDM) is an auxiliary operating mode that is used for system development. BDM is implemented in on-chip hardware and provides a full set of debug operations. Some BDM commands can be executed while the CPU is operating normally. Other BDM commands are firmware based, and require the BDM firmware to be enabled and active for execution.

In special single-chip mode, BDM is enabled and active immediately out of reset. BDM is available in all other operating modes, but must be enabled before it can be activated. BDM should not be used in special peripheral mode because of potential bus conflicts.

Once enabled, background mode can be made active by a serial command sent via the BKGD pin or execution of a CPU12 BGND instruction. While background mode is active, the CPU can interpret special debugging commands, and read and write CPU registers, peripheral registers, and locations in memory.

While BDM is active, the CPU executes code located in a small on-chip ROM mapped to addresses \$FF20 to \$FFFF, and BDM control registers are accessible at addresses \$FF00 to \$FF06. The BDM ROM replaces the regular system vectors while BDM is active. While BDM is active, the user memory from \$FF00 to \$FFFF is not in the map except through serial BDM commands.

**MODE** — Mode Register**\$000B**

	Bit 7	6	5	4	3	2	1	Bit 0	
	SMODN	MODB	MODA	ESTR	IVIS	0	EMD	EME	
RESET:	0	0	0	1	1	0	1	1	Special Single Chip
RESET:	0	0	1	1	1	0	1	1	Special Exp Nar
RESET:	0	1	0	1	1	0	1	1	Peripheral
RESET:	0	1	1	1	1	0	1	1	Special Exp Wide
RESET:	1	0	0	1	0	0	0	0	Normal Single Chip
RESET:	1	0	1	1	0	0	0	0	Normal Exp Nar
RESET:	1	1	1	1	0	0	0	0	Normal Exp Wide

MODE controls the MCU operating mode and various configuration options. This register is not in the map in peripheral mode

**SMODN, MODB, MODA** — Mode Select Special, B and A

These bits show the current operating mode and reflect the status of the BKGD, MODB and MODA input pins at the rising edge of reset.

Read anytime. SMODN may only be written if SMODN = 0 (in special modes) but the first write is ignored; MODB, MODA may be written once if SMODN = 1; anytime if SMODN = 0, except that special peripheral and reserved modes cannot be selected.

**ESTR** — E Clock Stretch Enable

Determines if the E Clock behaves as a simple free-running clock or as a bus control signal that is active only for external bus cycles.

0 = E never stretches (always free running).

1 = E stretches high during external access cycles and low during non-visible internal accesses.

Normal modes: write once; Special modes: write anytime, read anytime.

**IVIS** — Internal Visibility

This bit determines whether internal ADDR, DATA,  $R/\overline{W}$  and  $\overline{LSTRB}$  signals can be seen on the external bus during accesses to internal locations. In special narrow mode if this bit is set and EMD = 1 when an internal access occurs, the data appears wide on port C and port D. This allows for emulation. Visibility is not available when the part is operating in a single-chip mode.

0 = No visibility of internal bus operations on external bus.

1 = Internal bus operations are visible on external bus.

Normal modes: write once; Special modes: write anytime EXCEPT the first time. Read anytime.

## EMD — Emulate Port D

This bit only has meaning in special expanded narrow mode.

In expanded wide modes and special peripheral mode, PORTD, DDRD, KWIED and KWIFD are removed from the memory map regardless of the state of this bit.

In single-chip modes and normal expanded narrow mode, PORTD, DDRD, KWIED and KWIFD are in the memory map regardless of the state of this bit.

0 = PORTD, DDRD, KWIED and KWIFD are in the memory map.

1 = If in special expanded narrow mode; PORTD, DDRD, KWIED and KWIFD are removed from the memory map. Removing the registers from the map allows the user to emulate the function of these registers externally.

Normal modes: write once; Special modes: write anytime EXCEPT the first time. Read anytime.

## EME — Emulate Port E

In single-chip mode PORTE and DDRE are always in the map regardless of the state of this bit.

0 = PORTE and DDRE are in the memory map.

1 = If in an expanded mode, PORTE and DDRE are removed from the internal memory map. Removing the registers from the map allows the user to emulate the function of these registers externally.

Normal modes: write once; special modes: write anytime EXCEPT the first time. Read anytime.

## 6.3 Internal Resource Mapping

The internal register block, RAM, and EEPROM have default locations within the 64-Kbyte standard address space but may be reassigned to other locations during program execution by setting bits in mapping registers INITRG, INITRM, and INITEE. During normal operating modes these registers can be written once. It is advisable to explicitly establish these resource locations during the initialization phase of program execution, even if default values are chosen, in order to protect the registers from inadvertent modification later.

Writes to the mapping registers go into effect between the cycle that follows the write and the cycle after that. To assure that there are no unintended operations, a write to one of these registers should be followed with a NOP instruction.

If conflicts occur when mapping resources, the register block will take precedence over the other resources; RAM or EEPROM addresses occupied by the register block will not be available for storage. When active, BDM ROM takes precedence over other resources although a conflict between BDM ROM and register space is not possible. **Table 10** shows resource mapping precedence.

All address space not utilized by internal resources is, by default, external memory. The memory expansion module manages three memory overlay windows: program, data, and one extra page overlay. The size and location of the program and data overlay windows are fixed. One of two locations can be selected for the extra page (EPAGE).

**Table 10 Mapping Precedence**

Precedence	Resource
1	BDM ROM (if active)
2	Register Space
3	RAM
4	EEPROM
5	External Memory

### 6.3.1 Register Block Mapping

After reset the 512 byte register block resides at location \$0000 but can be reassigned to any 2-Kbyte boundary within the standard 64-Kbyte address space. Mapping of internal registers is controlled by five bits in the INITRG register. The register block occupies the first 512 bytes of the 2-Kbyte block.

**INITRG** — Initialization of Internal Register Position Register **\$0011**

	Bit 7	6	5	4	3	2	1	Bit 0
	REG15	REG14	REG13	REG12	REG11	0	0	0
RESET:	0	0	0	0	0	0	0	0

REG[15:11] — Internal register map position

These bits specify the upper five bits of the 16-bit registers address.

Normal modes: write once; special modes: write anytime. Read anytime.

### 6.3.2 RAM Mapping

The MC68HC812A4 has 1 Kbyte of fully static RAM that is used for storing instructions, variables, and temporary data during program execution. After reset, RAM addressing begins at location \$0800 but can be assigned to any 2-Kbyte boundary within the standard 64-Kbyte address space. Mapping of internal RAM is controlled by five bits in the INITRM register. The RAM array occupies the last 1 Kbyte of the 2-Kbyte block.

**INITRM** — Initialization of Internal RAM Position Register **\$0010**

	Bit 7	6	5	4	3	2	1	Bit 0
	RAM15	RAM14	RAM13	RAM12	RAM11	0	0	0
RESET:	0	0	0	0	1	0	0	0

RAM[15:11] — Internal RAM map position

These bits specify the upper five bits of the 16-bit RAM address.

Normal modes: write once; special modes: write anytime. Read anytime.

### 6.3.3 EEPROM Mapping

The MC68HC812A4 has 4 Kbytes of EEPROM which is activated by the EEON bit in the INITEE register.

Mapping of internal EEPROM is controlled by four bits in the INITEE register. After reset EEPROM address space begins at location \$1000 but can be mapped to any 4-Kbyte boundary within the standard 64-Kbyte address space.

**INITEE** — Initialization of Internal EEPROM Position Register **\$0012**

	Bit 7	6	5	4	3	2	1	Bit 0	
	EE15	EE14	EE13	EE12	0	0	0	EEON	
RESET:	0	0	0	1	0	0	0	1	Expand & Peripheral
RESET:	1	1	1	1	0	0	0	1	Single Chip

EE[15:12] — Internal EEPROM map position

These bits specify the upper four bits of the 16-bit EEPROM address.

Normal modes: write once; special modes: write anytime. Read anytime.

EEON — internal EEPROM On (Enabled)

This bit is forced to one in single-chip modes.

Read or write anytime.

0 = Removes the EEPROM from the map.

1 = Places the on-chip EEPROM in the memory map at the address selected by EE[15:12].

### 6.3.4 Expansion Address Mapping

Additional mapping controls are available that can be used in conjunction with memory expansion and chip selects.

To use memory expansion the part must be operated in one of the expanded modes. Sections of the standard 64-Kbyte memory map have memory expansion windows which allow more than 64 Kbytes to be addressed externally. Memory expansion on the MC68HC812A4 consists of three memory expansion windows and six address lines in addition to the existing standard 16 address lines. The memory expansion function reuses as many as six of the standard 16 address lines. Usage of chip selects will identify the source of the internal address.

All of the memory expansion windows have a fixed size and two of them have a fixed address location. The third has two selectable address locations.

**MISC** — Miscellaneous Mapping Control Register

**\$0013**

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

Normal modes: write once; Special modes: write anytime. Read anytime.

**EWDIR** — Extra Window Positioned in Direct Space

This bit is only valid in expanded modes. If the EWEN bit in the WINDEF register is cleared, then this bit has no meaning nor effect.

0 = If EWEN is set, then: a zero in this bit places the EPAGE at \$0400–\$07FF.

1 = If EWEN is set, then: a one in this bit places the EPAGE at \$0000–\$03FF.

**NDRC** — Narrow Data Bus for Register Chip Select Space

This function requires at least one of the chip selects CS[3:0] to be enabled. It effects the (external) 512-byte memory space.

0 = Makes the register-following chip select active space act as a full 16 bit data bus. If the narrow (8-bit) mode is being utilized this bit has no effect.

1 = Makes the register-following chip selects (2, 1, 0 and sometimes 3) active space [512-byte block] act the same as an 8-bit only external data bus (data only goes through port C externally). This allows 8-bit and 16-bit external memory devices to be mixed in a system.

## 6.4 Memory Maps

The following diagrams illustrate the memory map for each mode of operation immediately after reset.

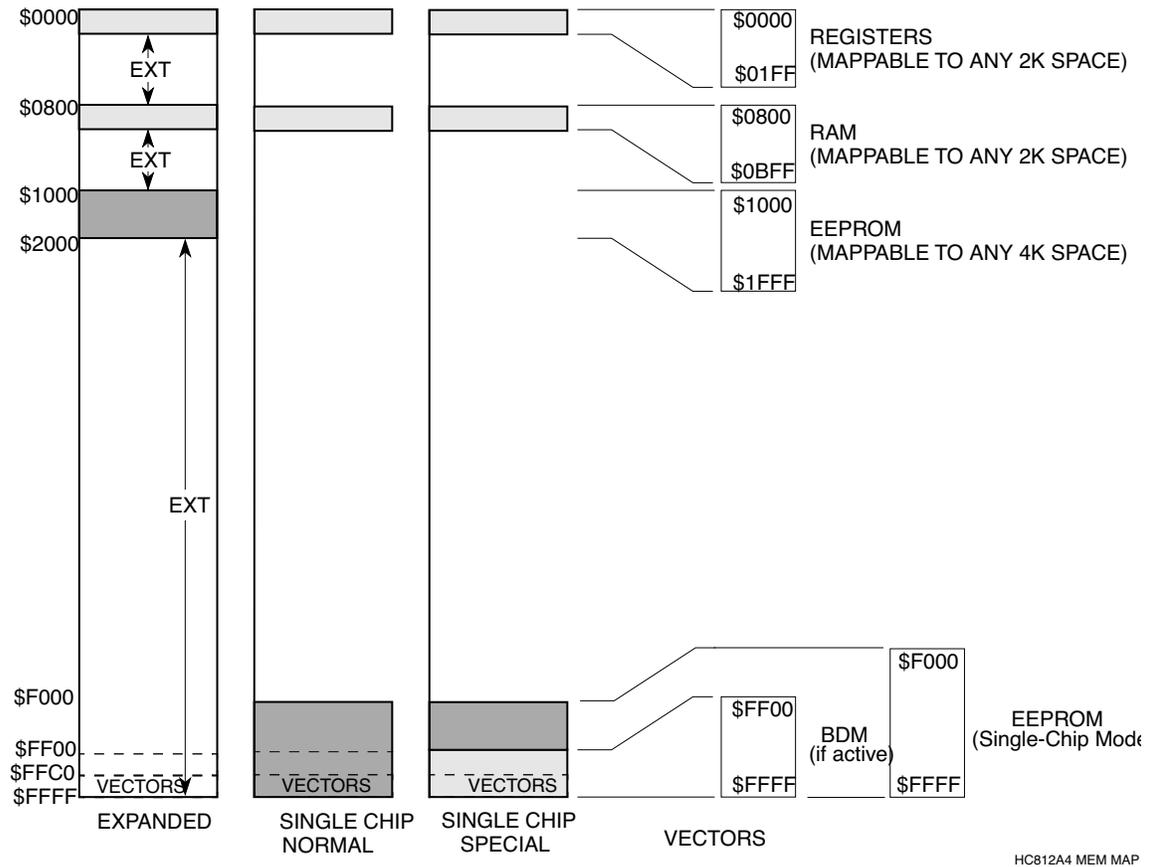


Figure 4 MC68HC812A4 Memory Map

## 7 EEPROM

The MC68HC812A4 EEPROM serves as a 4096-byte nonvolatile memory which can be used for frequently accessed static data or as fast access program code. Operating system kernels and standard subroutines would benefit from this feature.

The MC68HC812A4 EEPROM is arranged in a 16-bit configuration. The EEPROM array may be read as either bytes, aligned words or misaligned words. Access times is one bus cycle for byte and aligned word access and two bus cycles for misaligned word operations.

Programming is by byte or aligned word. Attempts to program or erase misaligned words will fail. Only the lower byte will be latched and programmed or erased. Programming and erasing of the user EEPROM can be done in all modes.

Each EEPROM byte or aligned word must be erased before programming. The EEPROM module supports byte, aligned word, row (32 bytes) or bulk erase, all using the internal charge pump. Bulk erasure of odd and even rows is also possible in test modes; the erased state is \$FF. The EEPROM module has hardware interlocks which protect stored data from corruption by accidentally enabling the program/erase voltage. Programming voltage is derived from the internal  $V_{DD}$  supply with an internal charge pump. The EEPROM has a minimum program/erase life of 10,000 cycles over the complete operating temperature range.

### 7.1 EEPROM Programmer's Model

The EEPROM module consists of two separately addressable sections. The first is a four-byte memory mapped control register block used for control, testing and configuration of the EEPROM array. The second section is the EEPROM array itself.

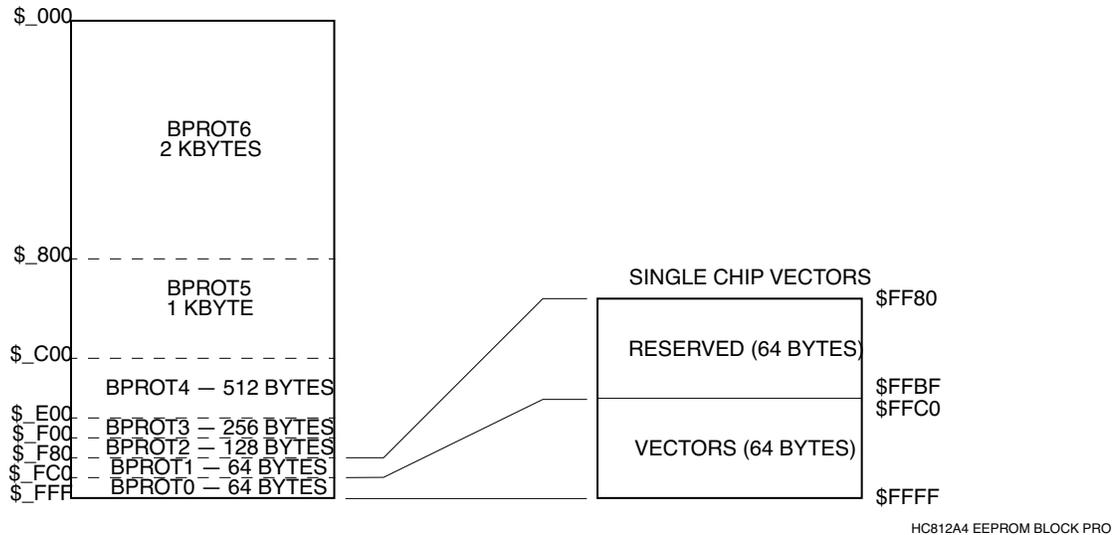
At reset, the four-byte register section starts at address \$00F0 and the EEPROM array is located from addresses \$1000 to \$1FFF (see **Figure 5**). For information on remapping the register block and EEPROM address space, refer to **6 Operating Modes and Resource Mapping**.

Read/write access to the memory array section can be enabled or disabled by the EEON control bit in the INITEE register. This feature allows the access of memory mapped resources that have lower priority than the EEPROM memory array. EEPROM control registers can be accessed and EEPROM locations may be programmed or erased regardless of the state of EEON.

Using the normal EEPROG control, it is possible to continue program/erase operations during WAIT. For lowest power consumption during WAIT, stop program/erase by turning off EEPGM.

If the STOP mode is entered during programming or erasing, program/erase voltage will be automatically turned off and the RC clock (if enabled) is stopped. However, the EEPGM control bit will remain set. When STOP mode is terminated, the program/erase voltage will be automatically turned back on if EEPGM is set.

At low bus frequencies, the RC clock must be turned on for program/erase.



**Figure 5 EEPROM Block Protect Mapping**

## 7.2 EEPROM Control Registers

**EEMCR** — EEPROM Module Configuration

**\$00F0**

	Bit 7	6	5	4	3	2	1	Bit 0
	1	1	1	1	1	EESWAI	PROTLCK	EERC
RESET:	1	1	1	1	1	1	0	0

**EESWAI** — EEPROM Stops in Wait Mode

- 0 = Module is not affected during wait mode
- 1 = Module ceases to be clocked during wait mode

This bit should be cleared if the wait mode vectors are mapped in the EEPROM array.

**PROTLCK** — Block Protect Write Lock

- 0 = Block protect bits and bulk erase protection bit can be written
- 1 = Block protect bits are locked

Read anytime. Write once in normal modes (SMODN = 1), set and clear any time in special modes (SMODN = 0).

**EERC** — EEPROM Charge Pump Clock

- 0 = System clock is used as clock source for the internal charge pump. Internal RC oscillator is stopped.
- 1 = Internal RC oscillator drives the charge pump. The RC oscillator is required when the system bus clock is lower than  $f_{PROG}$ .

Read and write anytime.

**EEPROT** — EEPROM Block Protect

**\$00F1**

	Bit 7	6	5	4	3	2	1	Bit 0
	1	BPROT6	BPROT5	BPROT4	BPROT3	BPROT2	BPROT1	BPROT0
RESET:	1	1	1	1	1	1	1	1

Prevents accidental writes to EEPROM. Read anytime. Write anytime if EEPGM = 0 and PROTLCK = 0.

### BPROT[6:0] — EEPROM Block Protection

0 = Associated EEPROM block can be programmed and erased.

1 = Associated EEPROM block is protected from being programmed and erased.

Cannot be modified while programming is taking place (EEPGM = 1).

**Table 11 4-Kbyte EEPROM Block Protection**

Bit Name	Block Protected	Block Size
BPROT6	\$1000 to \$17FF	2048 Bytes
BPROT5	\$1800 to \$1BFF	1024 Bytes
BPROT4	\$1C00 to \$1DFF	512 Bytes
BPROT3	\$1E00 to \$1EFF	256 Bytes
BPROT2	\$1F00 to \$1F7F	128 Bytes
BPROT1	\$1F80 to \$1FBF	64 Bytes
BPROT0	\$1FC0 to \$1FFF	64 Bytes

### EETST — EEPROM Test

**\$00F2**

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

Read anytime. Write in special modes only (SMODN = 0). These bits are used for test purposes only. In normal modes the bits are forced to zero.

#### EEODD — Odd Row Programming

0 = Odd row bulk programming/erasing is disabled.

1 = Bulk program/erase all odd rows.

#### EEVEN — Even Row Programming

0 = Even row bulk programming/erasing is disabled.

1 = Bulk program/erase all even rows.

#### MARG — Program and Erase Voltage Margin Test Enable

0 = Normal operation.

1 = Program and Erase Margin test.

This bit is used to evaluate the program/erase voltage margin.

#### EECPD — Charge Pump Disable

0 = Charge pump is turned on during program/erase.

1 = Disable charge pump.

#### EECPRD — Charge Pump Ramp Disable

Known to enhance write/erase endurance of EEPROM cells.

0 = Charge pump is turned on progressively during program/erase.

1 = Disable charge pump controlled ramp up.

#### EECPM — Charge Pump Monitor Enable

0 = Normal operation.

1 = Output the charge pump voltage on the  $\overline{\text{IRQ}}/V_{\text{PP}}$  pin.

### EEPROG — EEPROM Control

**\$00F3**

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

**BULKP — Bulk Erase Protection**

0 = EEPROM can be bulk erased.

1 = EEPROM is protected from being bulk or row erased.

Read anytime. Write anytime if EEPGM = 0 and PROTLCK = 0.

**BYTE — Byte and Aligned Word Erase**

0 = Bulk or row erase is enabled.

1 = One byte or one aligned word erase only.

Read anytime. Write anytime if EEPGM = 0.

**ROW — Row or Bulk Erase (when BYTE = 0)**

0 = Erase entire EEPROM array.

1 = Erase only one 32-byte row.

Read anytime. Write anytime if EEPGM = 0.

BYTE and ROW have no effect when ERASE = 0

**Table 12 Erase Selection**

Byte	Row	Block Size
0	0	Bulk erase entire EEPROM array
0	1	Row erase 32 bytes
1	0	Byte or aligned word erase
1	1	Byte or aligned word erase

If BYTE = 1 and test mode is not enabled, only the location specified by the address written to the programming latches will be erased. The operation will be a byte or an aligned word erase depending on the size of written data.

**ERASE — Erase Control**

0 = EEPROM configuration for programming.

1 = EEPROM configuration for erasure.

Read anytime. Write anytime if EEPGM = 0.

Configures the EEPROM for erasure or programming.

**EELAT — EEPROM Latch Control**

0 = EEPROM set up for normal reads.

1 = EEPROM address and data bus latches set up for programming or erasing.

Read anytime. Write anytime if EEPGM = 0.

**NOTE**

When EELAT is set, the entire EEPROM is unavailable for reads; therefore, no program residing in the EEPROM can be executed while attempting to program unused EEPROM space. Care should be taken that no references to the EEPROM are used while programming. Interrupts should be turned off if the vectors are in the EEPROM. Timing and any serial communications must be done with polling during the programming process.

BYTE, ROW, ERASE and EELAT bits can be written simultaneously or in any sequence.

**EEPGM — Program and Erase Enable**

0 = Disables program/erase voltage to EEPROM.

1 = Applies program/erase voltage to EEPROM.

The EEPGM bit can be set only after EELAT has been set. When EELAT and EEPGM are set simultaneously, EEPGM remains clear but EELAT is set.

The BULKP, BYTE, ROW, ERASE and EELAT bits cannot be changed when EEPGM is set. To complete a program or erase, two successive writes to clear EEPGM and EELAT bits are required before reading the programmed data. A write to an EEPROM location has no effect when EEPGM is set.

Latched address and data cannot be modified during program or erase.

A program or erase operation should follow the sequence below:

1. Write BYTE, ROW and ERASE to the desired value, write EELAT = 1
2. Write a byte or an aligned word to an EEPROM address
3. Write EEPGM = 1
4. Wait for programming ( $t_{\text{PROG}}$ ) or erase ( $t_{\text{erase}}$ ) delay time
5. Write EEPGM = 0
6. Write EELAT = 0

It is possible to program/erase more bytes or words without intermediate EEPROM reads by jumping from step 5 to step 2.

## 8 Memory Expansion and Chip Select

### 8.1 General Description of Memory Expansion

To use memory expansion the MC68HC812A4 must be operated in one of the expanded modes. Sections of the standard 64-Kbyte address space have memory expansion windows which allow an external address space larger than 64-Kbytes. Memory expansion consists of three memory expansion windows and six address lines which are used in addition to the standard 16 address lines.

The memory expansion function reuses as many as six of the standard 16 address lines. To do this, some of the upper address lines of internal addresses falling in an active window are overridden. Consequently, the address viewed externally may not match the internal address. Usage of chip selects will identify the source of the internal address for debugging and selection of the proper external devices.

All memory expansion windows have a fixed size and two have a fixed address location. The third has two selectable address locations. When an internal address falls into one of these active windows it is translated as shown in the table.

**Table 13 Memory Expansion Values (All Port G Assigned to Memory Expansion)**

Internal Address	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10
\$0000–\$03FF EWDIR <sup>1</sup> =1, EWEN=1	1	1	1	1	PEA17	PEA16	PEA15	PEA14	PEA13	PEA12	PEA11	PEA10
\$0000–\$03FF EWDIR OR EWEN =0	1	1	1	1	1	1	A15	A14	A13	A12	A11	A10
\$0400–\$07FF EWDIR =0, EWEN =1	1	1	1	1	PEA17	PEA16	PEA15	PEA14	PEA13	PEA12	PEA11	PEA10
\$0400–\$07FF EWDIR =1, EWEN =x OR EWDIR =x, EWEN =0	1	1	1	1	1	1	A15	A14	A13	A12	A11	A10
\$0800–\$6FFF	1	1	1	1	1	1	A15	A14	A13	A12	A11	A10
\$7000–\$7FFF DWEN = 1	1	1	PDA19	PDA18	PDA17	PDA16	PDA15	PDA14	PDA13	PDA12	A11	A10
\$7000–\$7FFF DWEN = 0	1	1	1	1	1	1	A15	A14	A13	A12	A11	A10
\$8000–\$BFFF PWEN = 1	PPA21	PPA20	PPA19	PPA18	PPA17	PPA16	PPA15	PPA14	A13	A12	A11	A10
\$8000–\$BFFF PWEN = 0	1	1	1	1	1	1	A15	A14	A13	A12	A11	A10
\$C000–\$FFFF	1	1	1	1	1	1	A15	A14	A13	A12	A11	A10

1. The EWDIR bit in the MISC register selects the E window address (1 = \$0000 to \$03FF including direct space, 0 = \$0400 to \$07FF).

Addresses ADDR[9:0] are not affected by memory expansion and are the same externally as they are internally. Addresses ADDR[21:16] are only generated by memory expansion and are individually enabled by software programmable control bits. If not enabled they may be used as general-purpose I/O. Addresses ADDR[15:10] can be the internal addresses or they can be modified by the memory expansion module. These are not available as general-purpose I/O in expanded modes.

### 8.2 Generation of Chip Selects

To use chip selects the MC68HC812A4 must be in one of the expanded modes. Each of the seven chip selects has an address space for which it is active — that is, when the current CPU address is in the range of that chip select it becomes active. Chip selects are generally used to reduce or eliminate external address decode logic. These active low signals usually are connected directly to the chip select pin of an external device.

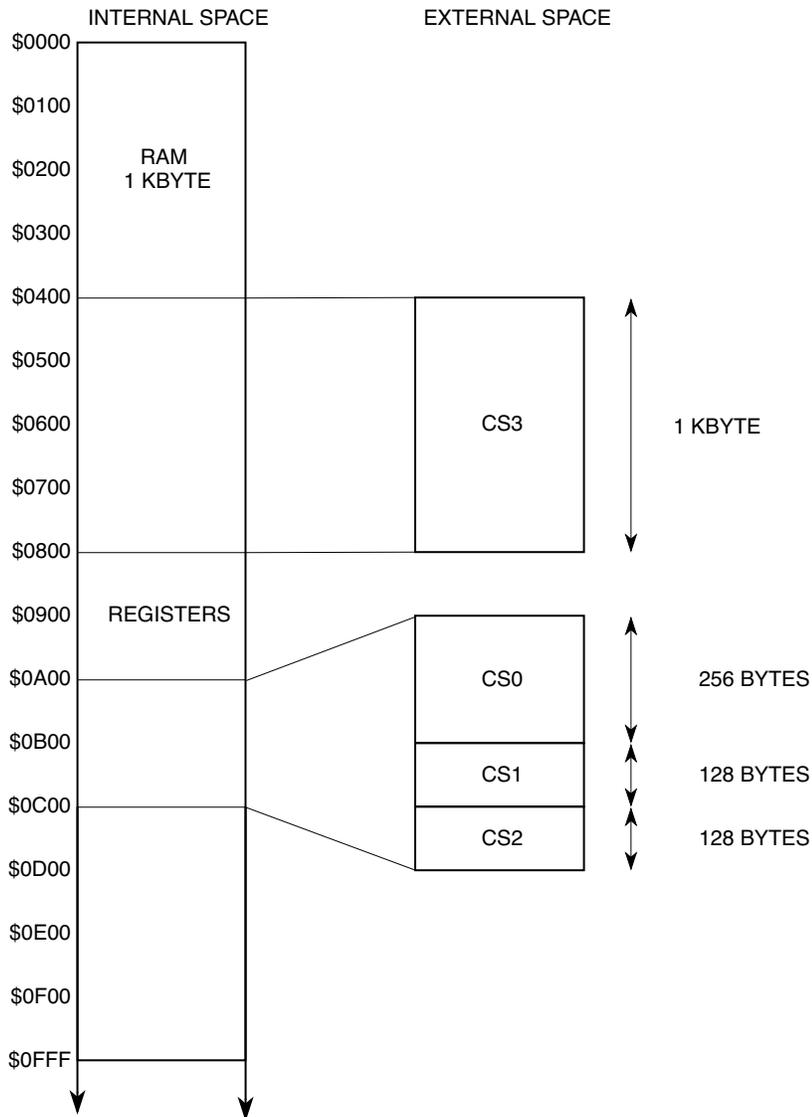
### 8.2.1 Chip Selects Independent of Memory Expansion

Three types of chip selects on the MC68HC812A4 are program memory chip selects, other memory chip selects and peripheral chip selects. Memory chip selects cover a medium to large address space. Peripheral chip selects (CS[3:0]) cover a small address space. The program memory chip select includes the vector space and is generally used with non-volatile memory. To start the user's program, the program chip select is designed to be active out of reset. This is the only chip select which has a functional difference from the others, so a small memory could use a peripheral chip select and a peripheral could use a memory chip select.

**Figure 6** shows peripheral chip selects in an expanded portion of the memory map. Chip selects CS[2:0] always map to the same 2-Kbyte block as the internal register space. The internal registers cover the first 512 bytes and these chip selects cover all or part of the 512 bytes following the register space blocking out a full 1-Kbyte space. CS3 can map with these other chip selects or be used in a 1-Kbyte space by itself which starts at either \$0000 or \$0400. CS3 can only be used for a 1-Kbyte space when it selects the E Page of memory expansion and E Page is active.

CS3 can be used with a 1-Kbyte space in systems not using memory expansion. However, it must be made to appear like memory expansion is in use. One of many possible configurations is the following:

- Select the desired 1-Kbyte space for EPAGE (EWDIR in MISC in the MMI)
- Write the EPAGE register with \$0000 if EWDIR is one or \$0001 if EWDIR is zero
- Designate all port G pins as I/O
- Enable EPAGE and CS3
- Make CS3 follow EPAGE



Register	Value	Meaning
INITRM	\$00	Assigns internal RAM to \$0000-\$0FFF
INITRG	\$08	Assigns register block to \$0800-\$09FF and register-following chip selects at \$0A00-\$0E
WINDEF	\$20	Enable EPAGE
MXAR	\$00	No port G lines assigned as extended address
CSCTL0	\$xF	Enables CS3, CS2, CS1, and CS0
CSCTL1	\$x8	Makes CS3 follow EPAGE
MISC	%0xxxxxx	Puts EPAGE at \$0400-\$07FF
EPAGE	\$01	Keeps the translated value of the upper addresses the same as it would have been before translation; not necessary if all external devices use chip selects.

HC812A4 CHIP SEL PART

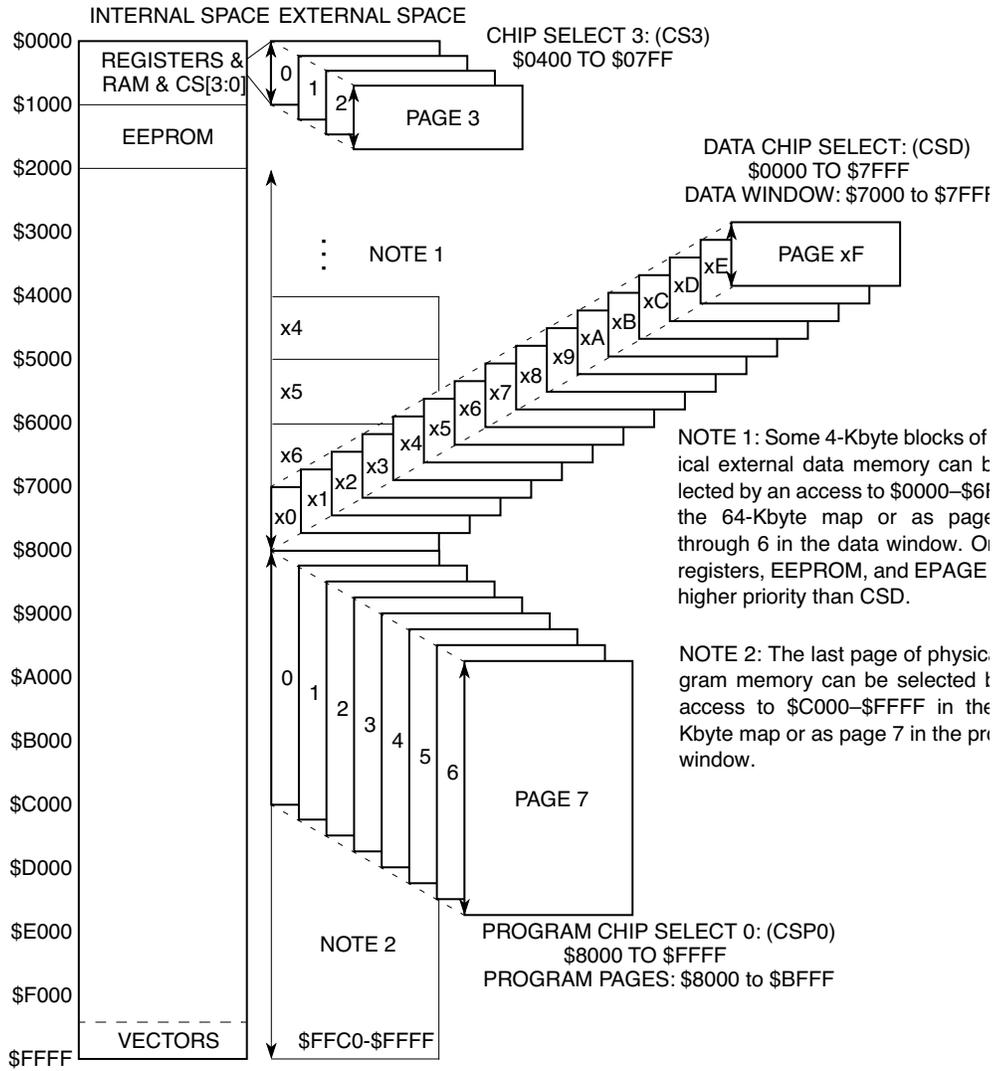
Figure 6 Chip Selects [3:0] (Partial Memory Map)

### 8.2.2 Chip Selects Used in Conjunction with Memory Expansion

Memory expansion and chip select functions can work independently, but systems requiring memory expansion perform better when chip selects are also used. For each memory expansion window there is a chip select (or two) designed to function with it.

**Figure 7** shows a memory expansion and chip select example using three chip selects. The program space consists of 128 Kbytes of addressable memory in eight 16-Kbyte pages. Page 7 is always accessible in the space from \$C000 to \$FFFF. The data space consists of 64 Kbytes of addressable memory in sixteen, 4-Kbyte pages. Unless CSD is used to select the external RAM, pages 0 through 6 appear in the \$0000 to \$6FFF space wherever there is no higher priority resource. The extra space consists of four, 1-Kbyte pages making 4 Kbytes of addressable memory.

If memory is increased to the maximum in this example, the program space will consist of 4-Mbytes of addressable space with 256, 16-Kbyte pages and page \$FF always available. The data space will be 1 Mbyte of addressable space with 256, 4-Kbyte pages and pages \$F0 to \$F6 mirrored to the \$0000 to \$6FFF space. The extra space will be 256 Kbytes of addressable space in 256 1-Kbyte pages.



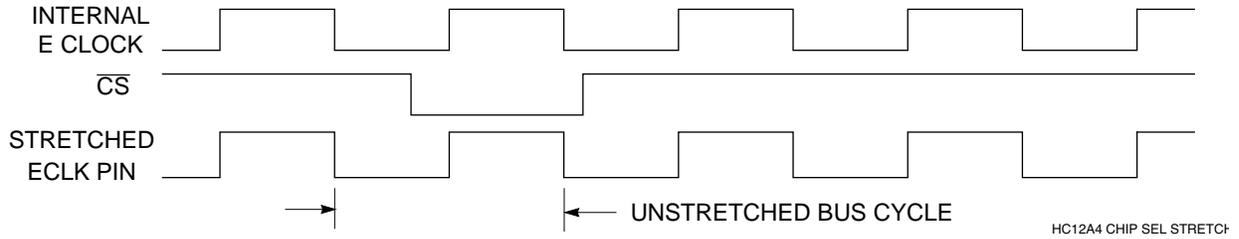
Register	Value	Meaning
WINDEF	\$E0	Enable EPAGE, DPAGE, PPAGE
MXAR	\$01	Port G bit 0 assigned as extended address ADDR16
CSCTL0	%00111xxx	Enables CSP0, CSD, and CS3
CSCTL1	\$18	Makes CSD follow \$0000-\$7FFF and CS3 select EPAGE
MISC	%0xxxxxx	Puts EPAGE at \$0400-\$09FF

HC812A4 MEM EXP CHIP SE

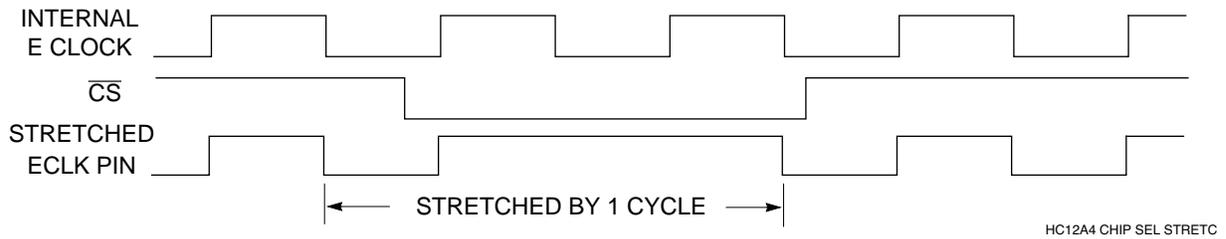
Figure 7 Memory Expansion and Chip Select Example

### 8.3 Chip Select Stretch

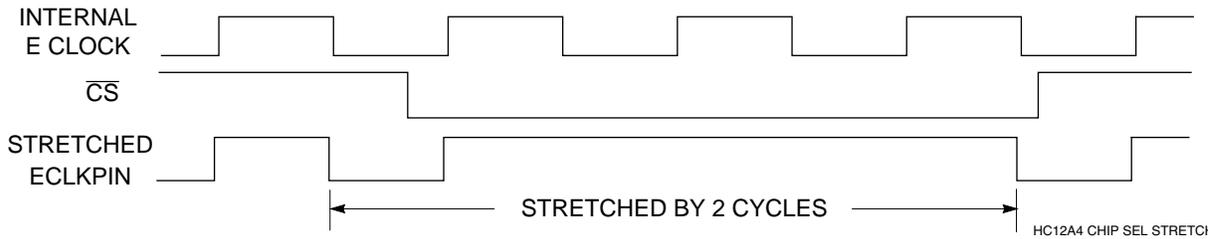
Each chip select can be chosen to stretch bus cycles associated with it. Stretch can be zero, one, two or three whole cycles added which allows interfacing to external devices which cannot meet full bus speed timing. The following figures show the waveforms for zero to three cycles of stretch.



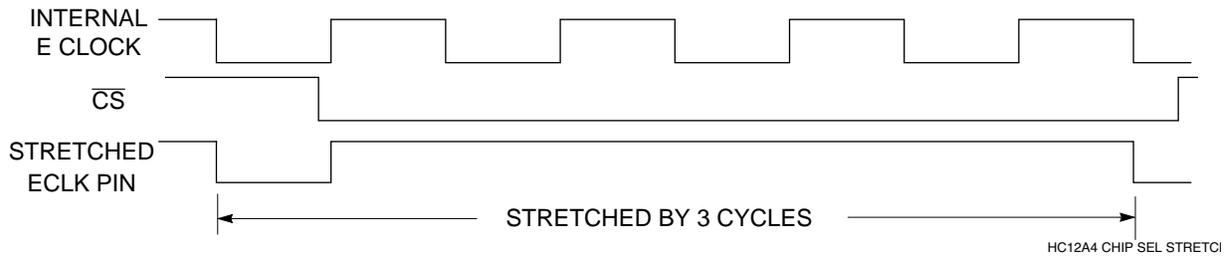
**Figure 8 Chip Select with No Stretch**



**Figure 9 Chip Select with One Cycle Stretch**



**Figure 10 Chip Select with Two Cycles Stretch**



**Figure 11 Chip Select with Three Cycles Stretch**

The external E clock may be the stretched E clock, the E clock, or no clock depending on the selection of control bits ESTR and IVIS in the MODE register and NECLK in the PEAR register.

## 8.4 Memory Expansion Registers

### PORTF — Port F Register

**\$0030**

	Bit 7	6	5	4	3	2	1	Bit 0
	0	Bit 6	5	4	3	2	1	Bit 0
RESET:	0	0	0	0	0	0	0	0
Alt. Pin Function	0	CSP1	CSP0	CSD	CS3	CS2	CS1	CS0

Seven port F pins are associated with chip selects. Any pin not used for chip select can be used as general-purpose I/O. All pins are pulled up when inputs (if pull-ups are enabled). Enabling a chip select overrides the associated data direction bit and port data bit.

Read and write anytime.

### PORTG — Port G Register

**\$0031**

	Bit 7	6	5	4	3	2	1	Bit 0
	0	0	Bit 5	4	3	2	1	Bit 0
RESET:	0	0	0	0	0	0	0	0
Alt. Pin Function	0	0	ADDR21	ADDR20	ADDR19	ADDR18	ADDR17	ADDR16

Six port G pins are associated with memory expansion. Any pin not used for memory expansion can be used as general-purpose I/O. All pins are pulled up when inputs (if pull-ups are enabled). Enabling a memory expansion address with the memory expansion assignment register overrides the associated data direction bit and port data bit.

Read and write anytime.

### DDRF — Port F Data Direction Register

**\$0032**

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

When port F is active, DDRF determines pin direction.

0 = Associated bit is an input.

1 = Associated bit is an output.

Read and write anytime.

### DDRG — Port G Data Direction Register

**\$0033**

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

When port G is active, DDRG determines pin direction.

0 = Associated bit is an input.

1 = Associated bit is an output.

Read and write anytime.

**DPAGE** — Data Page Register**\$0034**

	Bit 7	6	5	4	3	2	1	Bit 0
	PDA19	PDA18	PDA17	PDA16	PDA15	PDA14	PDA13	PDA12
RESET:	0	0	0	0	0	0	0	0

When enabled (DWEN = 1) the value in this register determines which of the 256 4-Kbyte pages is active in the data window. An access to the data page memory area (\$7000 to \$7FFF) forces the contents of DPAGE to address pins ADDR[15:12] and expansion address pins ADDR[19:16]. Bits ADDR20 and ADDR21 are forced to one (if enabled by MXAR). Data chip select (CSD) must be used in conjunction with this memory expansion window.

Read and write anytime.

**PPAGE** — Program Page Register**\$0035**

	Bit 7	6	5	4	3	2	1	Bit 0
	PPA21	PPA20	PPA19	PPA18	PPA17	PPA16	PPA15	PPA14
RESET:	0	0	0	0	0	0	0	0

When enabled (PWEN = 1) the value in this register determines which of the 256 16-Kbyte pages is active in the program window. An access to the program page memory area (\$8000 to \$BFFF) forces the contents of PPAGE to address pins ADDR[15:14] and expansion address pins ADDR[21:16]. At least one of the program chip selects (CSP0 or CSP1) must be used in conjunction with this memory expansion window. This register is used by the CALL and RTC instructions to facilitate automatic program flow changing between pages of program memory.

Read and write anytime.

**EPAGE** — Extra Page Register**\$0036**

	Bit 7	6	5	4	3	2	1	Bit 0
	PEA17	PEA16	PEA15	PEA14	PEA13	PEA12	PEA11	PEA10
RESET:	0	0	0	0	0	0	0	0

When enabled (EWEN = 1) the value in this register determines which of the 256 1-Kbyte pages is active in the extra window. An access to the extra page memory area forces the contents of EPAGE to address pins ADDR[15:10] and expansion address pins ADDR[16:17]. Address bits ADDR[21:18] are forced to one (if enabled by MXAR). Chip select 3 set to follow the extra page window (CS3 with CS3EP = 1) must be used in conjunction with this memory expansion window.

Read and write anytime.

**WINDEF** — Window Definition Register**\$0037**

	Bit 7	6	5	4	3	2	1	Bit 0
	DWEN	PWEN	EWEN	0	0	0	0	0
RESET:	0	0	0	0	0	0	0	0

Read and write anytime.

**DWEN** — Data Window Enable

0 = Disables DPAGE

1 = Enables paging of the data space (4 Kbytes: \$7000 – \$7FFF) via the DPAGE register

**PWEN** — Program Window Enable

0 = Disables PPAGE

1 = Enables paging of the program space (16 Kbytes: \$8000 – \$BFFF) via the PPAGE register

EWEN — Extra Window Enable

0 = Disables EPAGE

1 = Enables paging of the extra space (1 Kbyte) via the EPAGE register

**MXAR** — Memory Expansion Assignment Register

**\$0038**

	Bit 7	6	5	4	3	2	1	Bit 0
	0	0	A21E	A20E	A19E	A18E	A17E	A16E
RESET:	0	0	0	0	0	0	0	0

A21E, A20E, A19E, A18E, A17E, A16E — Selects the memory expansion pins ADDR[21:16].

0 = Selects general-purpose I/O for the associated bit function.

1 = Selects memory expansion for the associated bit function, overrides DDRG.

In single-chip modes these bits have no effect.

Read and write anytime.

## 8.5 Chip Selects

The chip selects are all active low. All pins in the associated port are pulled up when they are inputs and the PUPF bit in PUCR is set.

If memory expansion is used, chip selects should usually be used as well since some translated addresses can be confused with untranslated addresses that are not in an expansion window.

In single chip modes, enabling the chip select function does not affect the associated pins.

The block of register-following chip selects CS[3:0] allows many combinations including:

- 512-byte CS0
- 256-byte CS0 & 256-byte CS1
- 256-byte CS0 & 128-byte CS1 & 128-byte CS2
- 128-byte CS0 & 128-byte CS1 & 128-byte CS2 & 128-byte CS3

These register-following chip selects are available in the 512 byte space next to and higher in address than the 512 byte space which includes the registers. For example, if the registers are located at \$0800 to \$09FF, then these register-following chip selects are available in the space from \$0A00 to \$0BFF.

### 8.5.1 Chip Select Registers

**CSCTL0** — Chip Select Control Register 0

**\$003C**

	Bit 7	6	5	4	3	2	1	Bit 0
	0	CSP1E	CSP0E	CSDE	CS3E	CS2E	CS1E	CS0E
RESET:	0	0	1	0	0	0	0	0

Read and write: anytime.

Bits have no effect on the associated pin in single chip modes.

CSP1E — Chip Select Program 1 Enable

0 = Disables this chip select.

1 = Enables this chip select which covers the space \$8000 to \$FFFF or full map \$0000 to \$FFFF. Effectively selects the holes in the memory map. It can be used in conjunction with CSP0 to select between two 2-Mbyte devices based on address ADDR21.

CSP0E — Chip Select Program 0 Enable

0 = Disables this chip select.

1 = Enables this chip select which covers the program space \$8000 to \$FFFF.

**CSDE** — Chip Select Data Enable  
 0 = Disables this chip select.  
 1 = Enables this chip select which covers either \$0000 to \$7FFF (CSDHF = 1) or \$7000 to \$7FFF (CSDHF = 0).

**CS3E** — Chip Select 3 Enable  
 0 = Disables this chip select.  
 1 = Enables this chip select which covers a 128-byte space following the register space(\$x280–\$x2FF or \$xA80–\$xAFF). Alternately it can be active for accesses within the extra page window.

**CS2E** — Chip Select 2 Enable  
 0 = Disables this chip select.  
 1 = Enables this chip select which covers a 128-byte space following the register space (\$x380–\$x3FF or \$xB80–\$xBFF).

**CS1E** — Chip Select 1 Enable  
 0 = Disables this chip select.  
 1 = Enables this chip select which covers a 256-byte space following the register space (\$x300–\$x3FF or \$xB00–\$xBFF). CS2 and CS3 have a higher precedence and can override CS1 for a portion of this space.

**CS0E** — Chip Select 0 Enable  
 0 = Disables this chip select.  
 1 = Enables this chip select which covers a 512-byte space following the register space (\$x200–\$x3FF or \$xA00–\$xBFF). CS1, CS2, and CS3 have higher precedence and can override CS0 for portions of this space.

**CSCTL1** — Chip Select Control Register 1 **\$003D**

	Bit 7	6	5	4	3	2	1	Bit 0
	0	CSP1FL	CSPA21	CSDHF	CS3EP	0	0	0
RESET:	0	0	0	0	0	0	0	0

Read and write anytime.

**CSP1FL** — Program Chip Select 1 Covers Full Map  
 0 = If CSPA21 is cleared, chip select program 1 covers half the map, \$8000 to \$FFFF. If CSPA21 is set, this bit has no meaning nor effect.  
 1 = If CSPA21 is cleared, chip select program 1 covers the entire memory map. If CSPA21 is set, this bit has no meaning nor effect.

**CSPA21** — Program Chip Select Split Based on ADDR21  
 0 = CSP0 and CSP1 do not rely on ADDR21.  
 1 = Program chip selects are both active (if enabled) for space \$8000 to \$FFFF; CSP0 if ADDR21 is set and CSP1 if ADDR21 is cleared. This allows two 2-Mbyte memories to make up the 4-Mbyte addressable program space. Since ADDR21 is always one in the unpagged \$C000 to \$FFFF space, CSP0 is active in this space.

**CSDHF** — Data Chip Select Covers Half the Map  
 0 = Data chip select covers only \$7000 to \$7FFF (the optional data page window).  
 1 = Data chip select covers half the memory map (\$0000 to \$7FFF) including the optional data page window (\$7000 to \$7FFF).

**CS3EP** — Chip Select 3 Follows the Extra Page  
 0 = Chip select 3 includes only accesses to a 128-byte space following the register space.  
 1 = Chip select 3 follows accesses to the 1-Kbyte extra page (\$0400 to \$07FF or \$0000 to \$03FF). Any accesses to this window will cause the chip select to go active. (EWEN must be set to 1.)

## 8.5.2 CSSTR0–CSSTR1 Chip Select Stretch Registers 0 and 1

Each of the seven chip selects has a 2-bit field in this register which determines the amount of clock stretch for accesses in that chip select space.

Read and write anytime.

**CSSTR0** — Chip Select Stretch Register 0

**\$003E**

	Bit 7	6	5	4	3	2	1	Bit 0
	0	0	SRP1A	SRP1B	SRP0A	SRP0B	STRDA	STRDB
RESET:	0	0	1	1	1	1	1	1

**CSSTR1** — Chip Select Stretch Register 1

**\$003F**

	Bit 7	6	5	4	3	2	1	Bit 0
	STR3A	STR3B	STR2A	STR2B	STR1A	STR1B	STR0A	STR0B
RESET:	1	1	1	1	1	1	1	1

**Table 14 Stretch Bit Definition**

Stretch bit SxxxA	Stretch bit SxxxB	Number of E Clocks Stretched
0	0	0
0	1	1
1	0	2
1	1	3

## 8.6 Priority

Only one module or chip select may be selected at a time. If more than one module shares a space, only the highest priority module will be selected.

**Table 15 Module Priorities**

Priority	Module or Space
Highest	On-chip register space — 512 bytes fully blocked for registers though some of this space is unused
	BDM space (internal) when BDM is active this 256-byte block of registers and ROM appear at \$FFxx, cannot overlap RAM or registers
	On-chip RAM
	On-chip EEPROM (if enabled, EEON = 1)
	E space (external) <sup>1</sup> — 1 Kbyte at either \$0000 to \$03FF or \$0400 to \$07FF, may be used with “Extra” memory expansion and CS3
	CS space (external) <sup>1</sup> — 512 bytes following the 512-byte register space, may be used with CS[3:0]
	P space (external) <sup>1</sup> — 16 Kbyte fixed at \$8000 to \$BFFF, may be used with program memory expansion and CSP0 and/or CSP1
	D space (external) <sup>1</sup> — 4 Kbyte fixed at \$7000 to \$7FFF, may be used with data memory expansion and CSD or CSP1 (if set for full memory space) or the entire half of memory space \$0000–\$7FFF.
Lowest	Remaining external <sup>1</sup>

1. External spaces can only be accessed if the MCU is in expanded mode. Priorities of different external spaces affect chip selects and memory expansion.

Only one chip select will be active at any address. In the event that two or more chip selects cover the same address, only the highest priority chip select will be active. Chip selects have the following order of priority:

<b>Highest</b>							<b>Lowest</b>
	CS3	CS2	CS1	CS0	CSP0	CSD	CSP1

## 9 Resets and Interrupts

CPU12 exceptions include resets and interrupts. Each exception has an associated 16-bit vector, which points to the memory location where the routine that handles the exception is located. Vectors are stored in the upper 128 bytes of the standard 64-Kbyte address map.

The six highest vector addresses are used for resets and non-maskable interrupt sources. The remainder of the vectors are used for maskable interrupts, and all must be initialized to point to the address of the appropriate service routine.

### 9.1 Exception Priority

A hardware priority hierarchy determines which reset or interrupt is serviced first when simultaneous requests are made. Six sources are not maskable. The remaining sources are maskable, and any one of them can be given priority over other maskable interrupts.

The priorities of the non-maskable sources are:

1. POR or  $\overline{\text{RESET}}$  pin
2. Clock monitor reset
3. COP watchdog reset
4. Unimplemented instruction trap
5. Software interrupt instruction (SWI)
6.  $\overline{\text{XIRQ}}$  signal (if X bit in CCR = 0)

### 9.2 Maskable interrupts

Maskable interrupt sources include on-chip peripheral systems and external interrupt service requests. Interrupts from these sources are recognized when the global interrupt mask bit (I) in the CCR is cleared. The default state of the I bit out of reset is one, but it can be written at any time.

Interrupt sources are prioritized by default but any one maskable interrupt source may be assigned the highest priority by means of the HPRIO register. The relative priorities of the other sources remain the same.

An interrupt that is assigned highest priority is still subject to global masking by the I bit in the CCR, or by any associated local bits. Interrupt vectors are not affected by priority assignment. HPRIO can only be written while the I bit is set (interrupts inhibited). **Table 16** lists interrupt sources and vectors in default order of priority.

**Table 16 Interrupt Vector Map**

Vector Address	Interrupt Source	CCR Mask	Local Enable	HPRIO Value to Elevate
\$FFFE, \$FFFF	Reset	none	none	–
\$FFFC, \$FFFD	COP Clock Monitor Fail Reset	none	CME, FCME	–
\$FFFA, \$FFFB	COP Failure Reset	none	cop rate selected	–
\$FFF8, \$FFF9	Unimplemented Instruction Trap	none	none	–
\$FFF6, \$FFF7	SWI	none	none	–
\$FFF4, \$FFF5	XIRQ	X bit	none	–
\$FFF2, \$FFF3	IRQ or Key Wake Up D	I bit	IRQEN, KWIED[7:0]	\$F2
\$FFF0, \$FFF1	Real Time Interrupt	I bit	RTIE	\$F0
\$FFEE, \$FFEF	Timer Channel 0	I bit	TC0	\$EE
\$FFEC, \$FFED	Timer Channel 1	I bit	TC1	\$EC
\$FFEA, \$FFEB	Timer Channel 2	I bit	TC2	\$EA
\$FFE8, \$FFE9	Timer Channel 3	I bit	TC3	\$E8
\$FFE6, \$FFE7	Timer Channel 4	I bit	TC4	\$E6
\$FFE4, \$FFE5	Timer Channel 5	I bit	TC5	\$E4
\$FFE2, \$FFE3	Timer Channel 6	I bit	TC6	\$E2
\$FFE0, \$FFE1	Timer Channel 7	I bit	TC7	\$E0
\$FFDE, \$FFDF	Timer Overflow	I bit	TOI	\$DE
\$FFDC, \$FFDD	Pulse Accumulator Overflow	I bit	PAOVI	\$DC
\$FFDA, \$FFDB	Pulse Accumulator Input Edge	I bit	PAII	\$DA
\$FFD8, \$FFD9	SPI Serial Transfer Complete	I bit	SPI0E	\$D8
\$FFD6, \$FFD7	SCI 0	I bit	TIE0, TCIE0, RIE0, ILIE0	\$D6
\$FFD4, \$FFD5	SCI 1	I bit	TIE1, TCIE1, RIE1, ILIE1	\$D4
\$FFD2, \$FFD3	ATD	I bit	ADIE	\$D2
\$FFD0, \$FFD1	Key Wakeup J (stop wakeup)	I bit	KWIEJ[7:0]	\$D0
\$FFCE, \$FFCF	Key Wakeup H (stop wakeup)	I bit	KWIEH[7:0]	\$CE
\$FF80–\$FFCD	Reserved	I bit		\$80–\$CC

### 9.3 Interrupt Control and Priority Registers

**INTCR** — Interrupt Control Register

**\$001E**

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

**IRQE** —  $\overline{\text{IRQ}}$  Select Edge Sensitive Only

0 =  $\overline{\text{IRQ}}$  configured for low-level recognition.

1 =  $\overline{\text{IRQ}}$  configured to respond only to falling edges (on pin PE6/ $\overline{\text{IRQ}}$ ).

IRQE can be read anytime and written once in normal modes. In special modes, IRQE can be read anytime and written anytime, except the first write is ignored.

**IRQEN** — External  $\overline{\text{IRQ}}$  Enable

The  $\overline{\text{IRQ}}$  pin has an internal pull-up.

0 = External  $\overline{\text{IRQ}}$  pin and key wakeup D are disconnected from interrupt logic.

1 = External  $\overline{\text{IRQ}}$  pin and key wakeup D are connected to interrupt logic.

IRQEN can be read and written anytime in all modes.

## DLY — Enable Oscillator Start-Up Delay on Exit from STOP

The delay time of about 4096 cycles is based on the M clock rate chosen.

0 = No stabilization delay imposed on exit from STOP mode. A stable external oscillator must be supplied.

1 = Stabilization delay is imposed before processing resumes after STOP.

DLY can be read anytime and written once in normal modes. In special modes, DLY can be read and written anytime.

## HPRIO — Highest Priority I Interrupt

**\$001F**

	Bit 7	6	5	4	3	2	1	Bit 0
	1	1	PSEL5	PSEL4	PSEL3	PSEL2	PSEL1	0
RESET:	1	1	1	1	0	0	1	0

Write only if I mask in CCR = 1 (interrupts inhibited). Read anytime.

To give a maskable interrupt source highest priority, write the low byte of the vector address to the HPRIO register. For example, writing \$F0 to HPRIO would assign highest maskable interrupt priority to the real-time interrupt timer (\$FFF0). If an unimplemented vector address or a non-I-masked vector address (value higher than \$F2) is written, then  $\overline{\text{IRQ}}$  will be the default highest priority interrupt.

## 9.4 Resets

There are five possible sources of reset. Power-on reset (POR), external reset on the  $\overline{\text{RESET}}$  pin, and reset from the alternate reset pin (PE7/ARST) share the normal reset vector. The computer operating properly (COP) reset and the clock monitor reset each has a vector. Entry into reset is asynchronous and does not require a clock but the MCU cannot sequence out of reset without a system clock.

### 9.4.1 Power-On Reset

A positive transition on  $V_{DD}$  causes a power-on reset (POR). An external voltage level detector, or other external reset circuits, are the usual source of reset in a system. The POR circuit only initializes internal circuitry during cold starts and cannot be used to force a reset as system voltage drops.

### 9.4.2 External Reset

The CPU distinguishes between internal and external reset conditions by sensing whether the reset pin rises to a logic one in less than eight E-clock cycles after an internal device releases reset. When a reset condition is sensed, the  $\overline{\text{RESET}}$  pin is driven low by an internal device for about 16 E-clock cycles, then released. Eight E-clock cycles later it is sampled. If the pin is still held low, the CPU assumes that an external reset has occurred. If the pin is high, it indicates that the reset was initiated internally by either the COP system or the clock monitor.

To prevent a COP or clock monitor reset from being detected during an external reset, hold the reset pin low for at least 32 cycles. An external RC power-up delay circuit on the reset pin is not recommended — circuit charge time can cause the MCU to misinterpret the type of reset that has occurred.

### 9.4.3 COP Reset

The MCU includes a computer operating properly (COP) system to help protect against software failures. When COP is enabled, software must write \$55 and \$AA (in this order) to the COPRST register in order to keep a watchdog timer from timing out. Other instructions may be executed between these writes. A write of any value other than \$55 or \$AA or software failing to execute the sequence properly causes a COP reset to occur.

#### 9.4.4 Clock Monitor Reset

If clock frequency falls below a predetermined limit when the clock monitor is enabled, a reset occurs.

### 9.5 Effects of Reset

When a reset occurs, MCU registers and control bits are changed to known start-up states, as follows.

#### 9.5.1 Operating Mode and Memory Map

Operating mode and default memory mapping are determined by the states of the BGND, MODA, and MODB pins during reset. The SMODN, MODA, and MODB bits in the MODE register reflect the status of the mode-select inputs at the rising edge of reset. Operating mode and default maps can subsequently be changed according to strictly defined rules.

#### 9.5.2 Clock and Watchdog Control Logic

The COP watchdog system is enabled, with the CR[2:0] bits set for the longest duration time-out. The clock monitor is disabled. The RTIF flag is cleared and automatic hardware interrupts are masked. The rate control bits are cleared, and must be initialized before the RTI system is used. The DLY control bit is set to specify an oscillator start-up delay upon recovery from STOP mode.

#### 9.5.3 Interrupts

PSEL is initialized in the HPRIO register with the value \$F2, causing the external  $\overline{\text{IRQ}}$  pin to have the highest I-bit interrupt priority. The  $\overline{\text{IRQ}}$  pin is configured for level-sensitive operation (for wired-OR systems). However, the interrupt mask bits in the CPU12 CCR are set to mask interrupt requests.

#### 9.5.4 Parallel I/O

If the MCU comes out of reset in an expanded mode, port A and port B are used for the address bus, port C and port D (port C only, if in narrow mode) are used for a data bus, and port E pins are normally used to control the external bus (operation of port E pins can be affected by the PEAR register). If the MCU comes out of reset in a single-chip mode, all ports are configured as general-purpose high-impedance inputs (except in normal narrow expanded mode, PE3 is configured as an output driven high). Out of reset, port F (in expanded modes, PF5 is an active chip select), port G, port H, port J, port S, port T, and port AD are all configured as general-purpose inputs.

#### 9.5.5 Central Processing Unit

After reset, the CPU fetches a vector from the appropriate address, then begins executing instructions. The stack pointer and other CPU registers are indeterminate immediately after reset. The CCR X and I interrupt mask bits are set to mask any interrupt requests. The S bit is also set to inhibit the STOP instruction.

#### 9.5.6 Memory

After reset, the internal register block is located at \$0000–\$01FF and RAM is at \$0800–\$0BFF. EE-PROM is located at \$1000–\$1FFF in expanded modes and \$F000–\$FFFF in single-chip modes.

#### 9.5.7 Other Resources

The timer, serial communications interface (SCI), serial peripheral interface (SPI), and analog-to-digital converter (ATD) are off after reset.

## 9.6 Register Stacking

Once enabled, an interrupt request can be recognized at any time after the I bit in the CCR is cleared. When an interrupt service request is recognized, the CPU responds at the completion of the instruction being executed. Interrupt latency varies according to the number of cycles required to complete the instruction. Some of the longer instructions can be interrupted and will resume normally after servicing the interrupt.

When the CPU begins to service an interrupt, the instruction queue is cleared, the return address is calculated, and then it and the contents of the CPU registers are stacked as shown in **Table 17**.

**Table 17 Stacking Order on Entry to Interrupts**

Memory Location	CPU Registers
SP - 2	RTN <sub>H</sub> : RTN <sub>L</sub>
SP - 4	Y <sub>H</sub> : Y <sub>L</sub>
SP - 6	X <sub>H</sub> : X <sub>L</sub>
SP - 8	B : A
SP - 9	CCR

After the CCR is stacked, the I bit (and the X bit, if an  $\overline{XIRQ}$  interrupt service request is pending) is set to prevent other interrupts from disrupting the interrupt service routine. The interrupt vector for the highest priority source that was pending at the beginning of the interrupt sequence is fetched, and execution continues at the referenced location. At the end of the interrupt service routine, an RTI instruction restores the content of all registers from information on the stack, and normal program execution resumes.

## 10 Key Wakeups

The key wakeup feature of the MC68HC812A4 issues an interrupt that will wake up the CPU when it is in the STOP or WAIT mode. Three ports are associated with the key wakeup function: port D, port H, and port J. Port D and port H wakeups are triggered with a falling signal edge. Port J key wakeups have a selectable falling or rising signal edge as the active edge. For each pin which has an interrupt enabled there is a path to the interrupt request signal which has no clocked devices when the part is in stop mode. This allows an active edge to bring the part out of stop.

Default register addresses, as established after reset, are indicated in the following descriptions. For information on remapping the register block, refer to **6 Operating Modes and Resource Mapping**.

### 10.1 Key Wakeup Registers

#### PORTD — Port D Register

**\$0005**

	Bit 7	6	5	4	3	2	1	Bit 0
	PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0
RESET:	0	0	0	0	0	0	0	0
Alt. Pin Function	KWD7	KWD6	KWD5	KWD4	KWD3	KWD2	KWD1	KWD0

This register is not in the map in wide expanded modes or in special expanded narrow mode with MODE register bit EMD set.

An interrupt is generated when a bit in the KWIFD register and its corresponding KWIED bit are both set. These bits correspond to the pins of port D. All eight bits/pins share the same interrupt vector and can wake the CPU when it is in STOP or WAIT mode. Key wakeups can be used with the pins configured as inputs or outputs.

Key wakeup port D shares a vector and control bit with  $\overline{IRQ}$ . IRQEN must be set for key wakeup interrupts to signal the CPU.

#### DDRD — Port D Data Direction Register

**\$0007**

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

This register is not in the map in wide expanded modes or in special expanded narrow mode with MODE register bit EMD set.

Data direction register D is associated with port D and designates each pin as an input or output.

Read and write anytime.

#### DDRD[7:0] — Data Direction Port D

0 = Associated pin is an input

1 = Associated pin is an output

#### KWIED — Key Wakeup Port D Interrupt Enable Register

**\$0020**

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

This register is not in the map in wide expanded modes and in special expanded narrow mode with MODE register bit EMD set.

Read and write anytime.

**KWIED[7:0]** — Key Wakeup Port D Interrupt Enables  
 0 = Interrupt for the associated bit is disabled  
 1 = Interrupt for the associated bit is enabled

**KWIFD** — Key Wakeup Port D Flag Register

**\$0021**

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

Each flag is set by a falling edge on its associated input pin. To clear the flag, write one to the corresponding bit in KWIFD.

This register is not in the map in wide expanded modes or in special expanded narrow mode with MODE register bit EMD set.

Read and write anytime

**KWIFD[7:0]** — Key Wakeup Port D Flags

0 = Falling edge on the associated bit has not occurred  
 1 = Falling edge on the associated bit has occurred (an interrupt will occur if the associated enable bit is set).

**PORTH** — Port H Register

**\$0024**

	Bit 7	6	5	4	3	2	1	Bit 0
RESET:	0	0	0	0	0	0	0	0
Alt. Pin Function	KWH7	KWH6	KWH5	KWH4	KWH3	KWH2	KWH1	KWH0

Port H is associated with key wakeup H. Key wakeups can be used with the pins designated as inputs or outputs. DDRH determines whether each pin is an input or output.

Read and write anytime.

**DDRH** — Port H Data Direction Register

**\$0025**

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

Data direction register H is associated with port H and designates each pin as an input or output.

Read and write anytime.

**DDRH[7:0]** — Data Direction Port H

0 = Associated pin is an input  
 1 = Associated pin is an output

**KWIEH** — Key Wakeup Port H Interrupt Enable Register

**\$0026**

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

An interrupt is generated when a bit in the KWIFH register and its corresponding KWIEH bit are both set. These bits correspond to the pins of port H.

**KWIEH[7:0]** — Key Wakeup Port H Interrupt Enables

0 = Interrupt for the associated bit is disabled  
 1 = Interrupt for the associated bit is enabled

**KWIFH** — Key Wakeup Port H Flag Register**\$0027**

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

Each flag is set by a falling edge on its associated input pin. To clear the flag, write one to the corresponding bit in KWIFH.

Read and write anytime.

**KWIFH[7:0]** — Key Wakeup Port H Flags

0 = Falling edge on the associated bit has not occurred

1 = Falling edge on the associated bit has occurred (an interrupt will occur if the associated enable bit is set)

**PORTJ** — Port J Register**\$0028**

	Bit 7	6	5	4	3	2	1	Bit 0
RESET:	0	0	0	0	0	0	0	0
Alt. Pin Function	KWJ7	KWJ6	KWJ5	KWJ4	KWJ3	KWJ2	KWJ1	KWJ0

Port J is associated with key wakeup J. Key wakeups can be used with the pins designated as inputs or outputs. DDRJ determines whether each pin is an input or output.

Read and write anytime.

**DDRJ** — Port J Data Direction Register**\$0029**

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

Determines direction of each port J pin.

**DDRJ[7:0]** — Data direction Port J

0 = Associated pin is an input

1 = Associated pin is an output

**KWIEJ** — Key Wakeup Port J Interrupt Enable Register**\$002A**

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

An interrupt is generated when a bit in the KWIFJ register and its corresponding KWIEJ bit are both set. These bits correspond to the pins of port J. All eight bits/pins share the same interrupt vector.

Read and write anytime.

**KWIEJ[7:0]** — Key Wakeup Port J Interrupt Enables

0 = Interrupt for the associated bit is disabled

1 = Interrupt for the associated bit is enabled

**KWIFJ** — Key Wakeup Port J Flag Register**\$002B**

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

Each flag gets set by an active edge on the associated input pin. This could be a rising or falling edge based on the state of the KPOLJ register. To clear the flag, write one to the corresponding bit in KWIFJ.

Initialize this register after initializing KPOLJ so that illegal flags can be cleared.

Read and write anytime.

**KWIFJ[7:0]** — Key Wakeup Port J Flags

0 = An active edge on the associated bit has not occurred

1 = An active edge on the associated bit has occurred (an interrupt will occur if the associated enable bit is set)

**KPOLJ** — Key Wakeup Port J Polarity Register

**\$002C**

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

Read and write anytime.

It is best to clear the flags after initializing this register because changing the polarity of a bit can cause the associated flag to go set.

**KPOLJ[7:0]** — Key Wakeup Port J Polarity Select

0 = Falling edge on the associated port J pin sets the associated flag bit in the KWIFJ register

1 = Rising edge on the associated port J pin sets the associated flag bit in the KWIFJ register

**PUPSJ** — Key Wakeup Port J Pull-Up/Pulldown Select Register

**\$002D**

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

Each bit in the register corresponds to a port J pin. Each bit selects a pull-up or pulldown device for the associated port J pin. The pull-up or pulldown will only be active if enabled by the PULEJ register.

PUPSJ should be initialized before enabling the pull-ups/pulldowns (PUPEJ).

Read and write anytime.

**PUPSJ[7:0]** — Key Wakeup Port J Pull-Up/Pulldown Select

0 = Pulldown is selected for the associated port J pin

1 = Pull-up is selected for the associated port J pin

**PULEJ** — Key Wakeup Port J Pull-Up/Pulldown Enable Register

**\$002E**

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

Each bit in the register corresponds to a port J pin. If a pin is configured as an input, each bit enables an active pull-up or pulldown device. PUPSJ selects whether a pull-up or a pulldown is the active device.

Read and write anytime.

**PULEJ[7:0]** — Key Wakeup Port J Pull-Up/Pulldown Enable

0 = Associated port J pin has no pull-up/pulldown device.

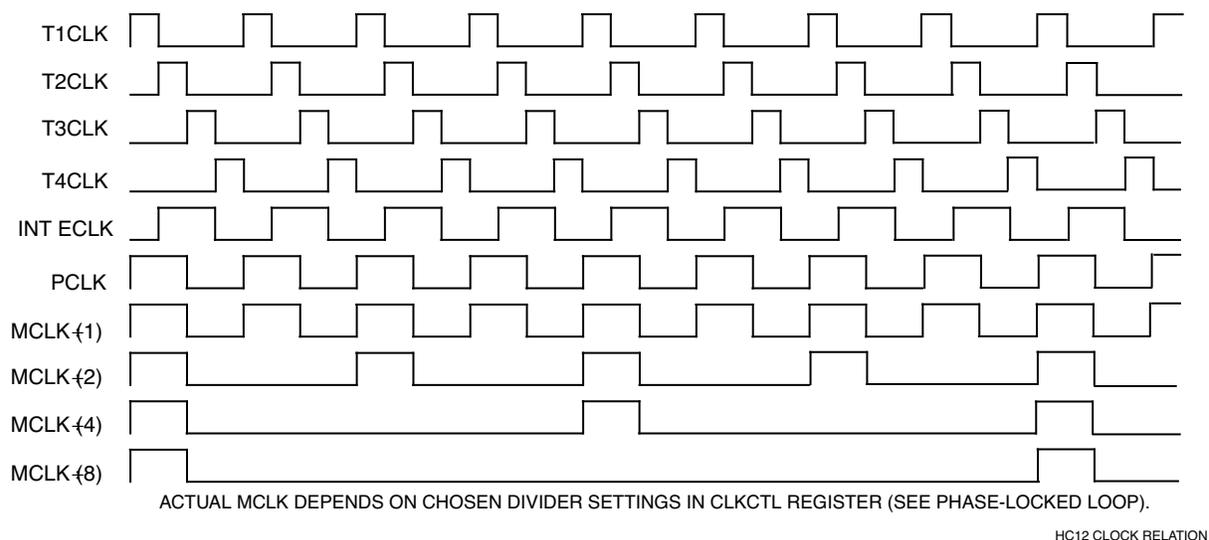
1 = Selected pull-up/pulldown device for the associated port J pin is enabled if it is an input.

## 11 Clock Functions

Clock generation circuitry generates the internal and external E-clock signals as well as internal clock signals used by the CPU and on-chip peripherals. A clock monitor circuit, a computer operating properly (COP) watchdog circuit, and a periodic interrupt circuit are also incorporated into the MC68HC812A4.

### 11.1 Clock Sources

A compatible external clock signal can be applied to the EXTAL pin or the MCU can generate a clock signal using an on-chip oscillator circuit and an external crystal or ceramic resonator. The MCU uses four types of internal clock signals derived from the primary clock signal: T clocks, E clock, P clock and M clock. The T clocks are used by the CPU. The E and P clocks are used by the bus interfaces, BDM, SPI, and ATD. The M clock drives on-chip modules such as the timer chain, SCI, RTI, COP, and restart-from-stop delay time. **Figure 12** shows clock timing relationships. Four bits in the CLKCTL register control the base clock and M-clock divide selection ( $\div 1$ ,  $\div 2$ ,  $\div 4$ , and  $\div 8$  are selectable).



**Figure 12 Internal Clock Relationships**

### 11.2 Computer Operating Properly (COP)

The COP or watchdog timer is an added check that a program is running and sequencing properly. When the COP is being used, software is responsible for keeping a free running watchdog timer from timing out. If the watchdog timer times out it is an indication that the software is no longer being executed in the intended sequence; thus a system reset is initiated. Three control bits allow selection of seven COP time-out periods. When COP is enabled, sometime during the selected period the program must write \$55 and \$AA (in this order) to the COPRST register. If the program fails to do this the part will reset. If any value other than \$55 or \$AA is written, the part is reset.

### 11.3 Real-Time Interrupt

There is a real time (periodic) interrupt available to the user. This interrupt will occur at one of seven selected rates. An interrupt flag and an interrupt enable bit are associated with this function. There are three bits for the rate select.

### 11.4 Clock Monitor

The clock monitor circuit is based on an internal resistor-capacitor (RC) time delay. If no MCU clock edges are detected within this RC time delay, the clock monitor can optionally generate a system reset.

The clock monitor function is enabled/disabled by the CME control bit in the COPCTL register. This time-out is based on an RC delay so that the clock monitor can operate without any MCU clocks.

Clock monitor time-outs are shown in **Table 18**.

**Table 18 Clock Monitor Time-Outs**

Supply	Range
5 V +/- 10%	2-20 $\mu$ S
3 V +/- 10%	5-100 $\mu$ S

### 11.5 Clock Function Registers

All register addresses shown reflect the reset state. Registers may be mapped to any 2-Kbyte space.

**RTICTL** — Real-Time Interrupt Control Register

**\$0014**

	Bit 7	6	5	4	3	2	1	Bit 0
	RTIE	RSWAI	RSBCK	0	RTBYP	RTR2	RTR1	RTR0
RESET:	0	0	0	0	0	0	0	0

**RTIE** — Real Time Interrupt Enable

Read and write anytime.

- 0 = Interrupt requests from RTI are disabled.
- 1 = Interrupt will be requested whenever RTIF is set.

**RSWAI** — RTI and COP Stop While in Wait

Write once in normal modes, anytime in special modes. Read anytime.

- 0 = Allows the RTI and COP to continue running in wait.
- 1 = Disables both the RTI and COP whenever the part goes into Wait.

**RSBCK** — RTI and COP Stop While in Background Debug Mode

Write once in normal modes, anytime in special modes. Read anytime.

- 0 = Allows the RTI and COP to continue running while in background mode.
- 1 = Disables both the RTI and COP whenever the part is in background mode. This is useful for emulation.

**RTBYP** — Real Time Interrupt Divider Chain Bypass

Write not allowed in normal modes, anytime in special modes. Read anytime.

- 0 = Divider chain functions normally.
- 1 = Divider chain is bypassed, allows faster testing (the divider chain is normally  $M$  divided by  $2^{13}$ , when bypassed becomes  $M$  divided by 4).

**RTR2, RTR1, RTR0** — Real-Time Interrupt Rate Select

Read and write anytime.

Rate select for real-time interrupt. The clock used for this module is the Module (M) clock.

**Table 19 Real Time Interrupt Rates**

RTR2	RTR1	RTR0	Divide M By:	Time-Out Period M = 4.0 MHz	Time-Out Period M = 8.0 MHz
0	0	0	OFF	OFF	OFF
0	0	1	2 <sup>13</sup>	2.048 ms	1.024 ms
0	1	0	2 <sup>14</sup>	4.096 ms	2.048 ms
0	1	1	2 <sup>15</sup>	8.196 ms	4.096 ms
1	0	0	2 <sup>16</sup>	16.384 ms	8.196 ms
1	0	1	2 <sup>17</sup>	32.768 ms	16.384 ms
1	1	0	2 <sup>18</sup>	65.536 ms	32.768 ms
1	1	1	2 <sup>19</sup>	131.72 ms	65.536 ms

**RTIFLG** — Real Time Interrupt Flag Register

**\$0015**

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

**RTIF** — Real Time Interrupt Flag

This bit is cleared automatically by a write to this register with this bit set.

0 = Time-out has not yet occurred.

1 = Set when the time-out period is met.

**COPCTL** — COP Control Register

**\$0016**

	Bit 7	6	5	4	3	2	1	Bit 0	
CME	FCME	FCM	FCOP	DISR	CR2	CR1	CR0		
RESET:	0	0	0	0	0	1	1	1	Normal
RESET:	0	0	0	0	1	1	1	1	Special

**CME** — Clock Monitor Enable

Read and write anytime.

If FCME is set, this bit has no meaning nor effect.

0 = Clock monitor is disabled. Slow clocks and stop instruction may be used.

1 = Slow or stopped clocks (including the stop instruction) will cause a clock reset sequence.

**FCME** — Force Clock Monitor Enable

Write once in normal modes, anytime in special modes. Read anytime.

In normal modes, when this bit is set, the clock monitor function cannot be disabled until a reset occurs.

0 = Clock monitor follows the state of the CME bit.

1 = Slow or stopped clocks will cause a clock reset sequence.

In order to use both STOP and clock monitor, the CME bit should be cleared prior to executing a STOP instruction and set after recovery from STOP. If you plan on using STOP always keep FCME = 0.

**FCM** — Force Clock Monitor Reset

Writes are not allowed in normal modes, anytime in special modes. Read anytime.

If DISR is set, this bit has no effect.

0 = Normal operation.

1 = Force a clock monitor reset (if clock monitor is enabled).

### FCOP — Force COP Watchdog Reset

Writes are allowed anytime in special modes but are not allowed in normal modes. Read anytime.

If DISR is set, this bit has no effect.

0 = Normal operation.

1 = Force a COP reset (if COP is enabled).

### DISR — Disable Resets from COP Watchdog and Clock Monitor

Writes are allowed anytime in special modes but are not allowed in normal modes. Read anytime.

0 = Normal operation.

1 = Regardless of other control bit states, COP and clock monitor will not generate a system reset.

### CR2, CR1, CR0 — COP Watchdog Timer Rate select bits

The COP system is driven by a constant frequency of  $M/2^{13}$ . These bits specify an additional division factor to arrive at the COP time-out rate (the clock used for this module is the M clock).

Write once in normal modes, anytime in special modes. Read anytime.

**Table 20 COP Watchdog Rates**

CR2	CR1	CR0	Divide M By:	At M = 4.0 MHz Time-Out -0/+2.048 ms	At M = 8.0 MHz Time-Out -0/+1.024 ms
0	0	0	OFF	OFF	OFF
0	0	1	$2^{13}$	2.048 ms	1.024 ms
0	1	0	$2^{15}$	8.1920 ms	4.096 ms
0	1	1	$2^{17}$	32.768 ms	16.384 ms
1	0	0	$2^{19}$	131.072 ms	65.536 ms
1	0	1	$2^{21}$	524.288 ms	262.144 ms
1	1	0	$2^{22}$	1.048 s	524.288 ms
1	1	1	$2^{23}$	2.097 s	1.048576 s

### COPRST — Arm/Reset COP Timer Register

**\$0017**

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

Always reads \$00.

Writing \$55 to this address is the first step of the COP watchdog sequence.

Writing \$AA to this address is the second step of the COP watchdog sequence. Other instructions may be executed between these writes but both must be completed in the correct order prior to time-out to avoid a watchdog reset. Writing anything other than \$55 or \$AA causes a COP reset to occur.

## 11.6 Clock Divider Chains

Figure 13, Figure 14, Figure 15, and Figure 16 summarize the clock divider chains for the various peripherals on the MC68HC812A4.

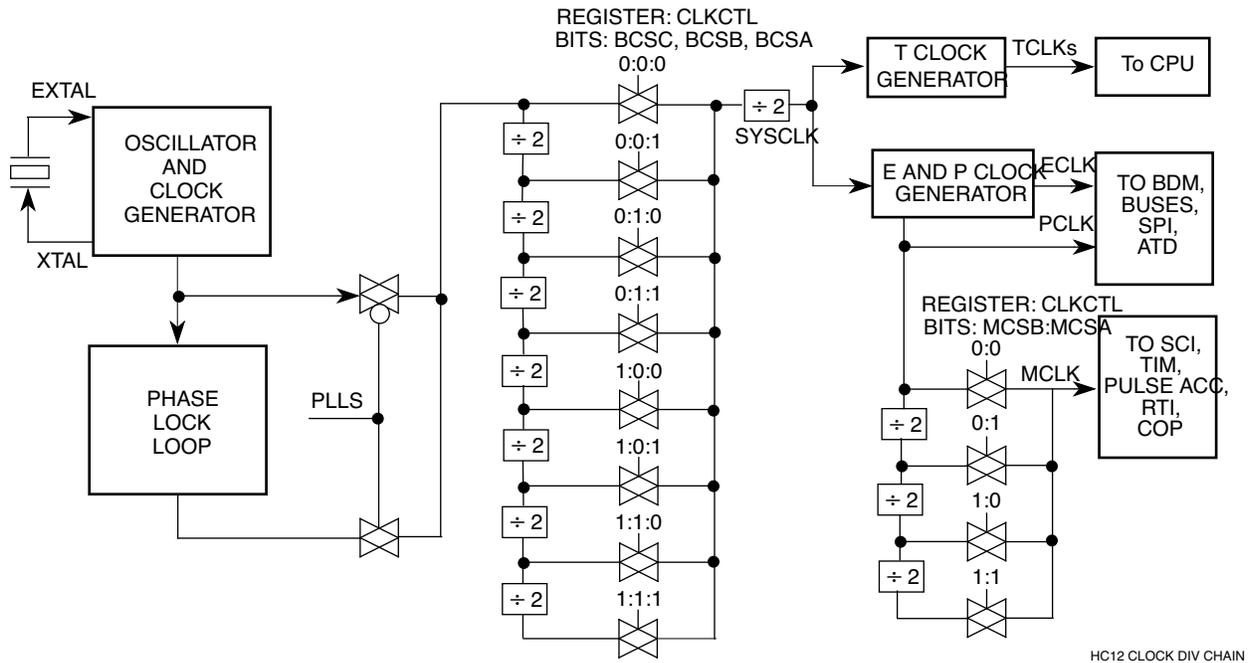


Figure 13 Clock Divider Chain

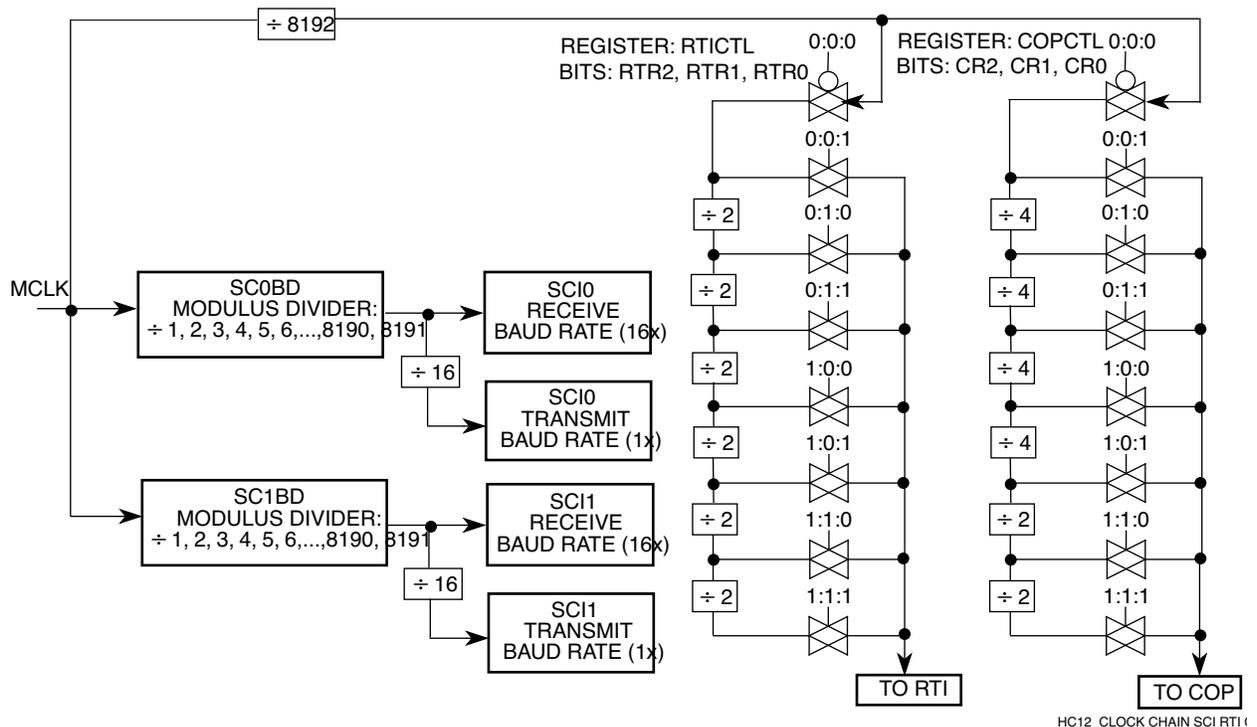


Figure 14 Clock Chain for SCI0, SCI1, RTI, COP

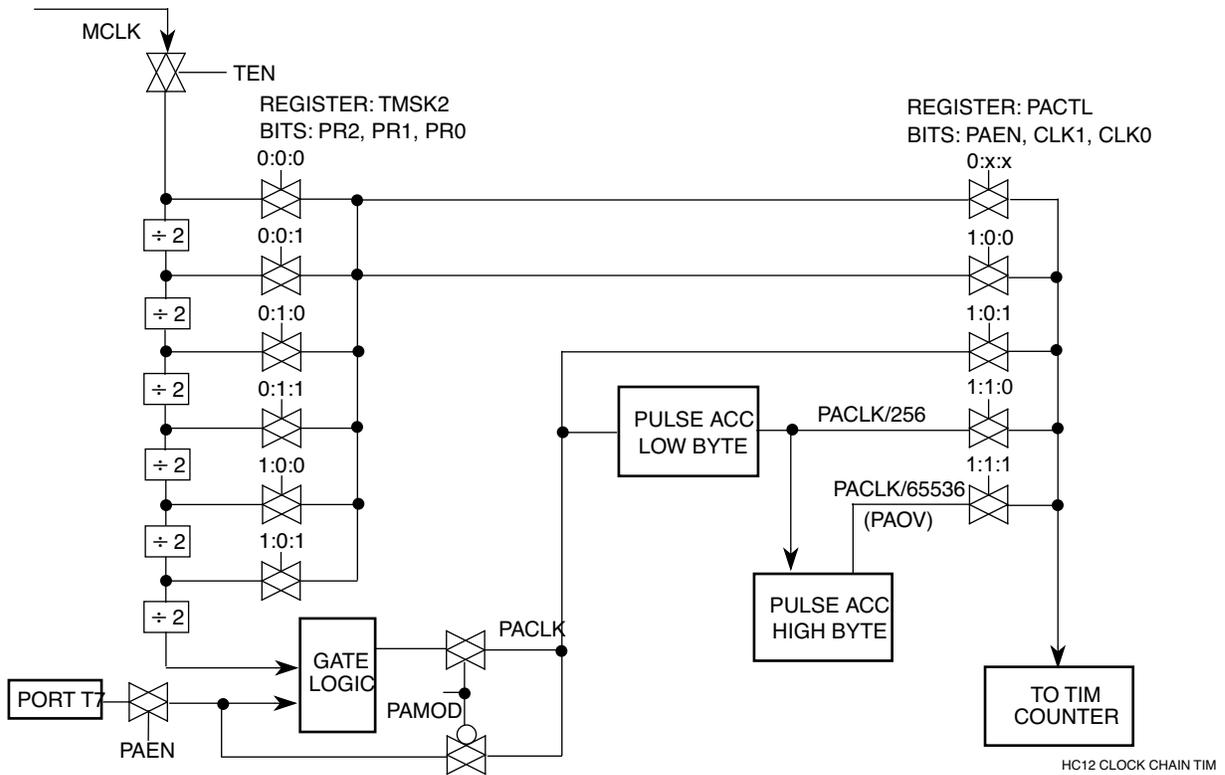


Figure 15 Clock Chain for TIM

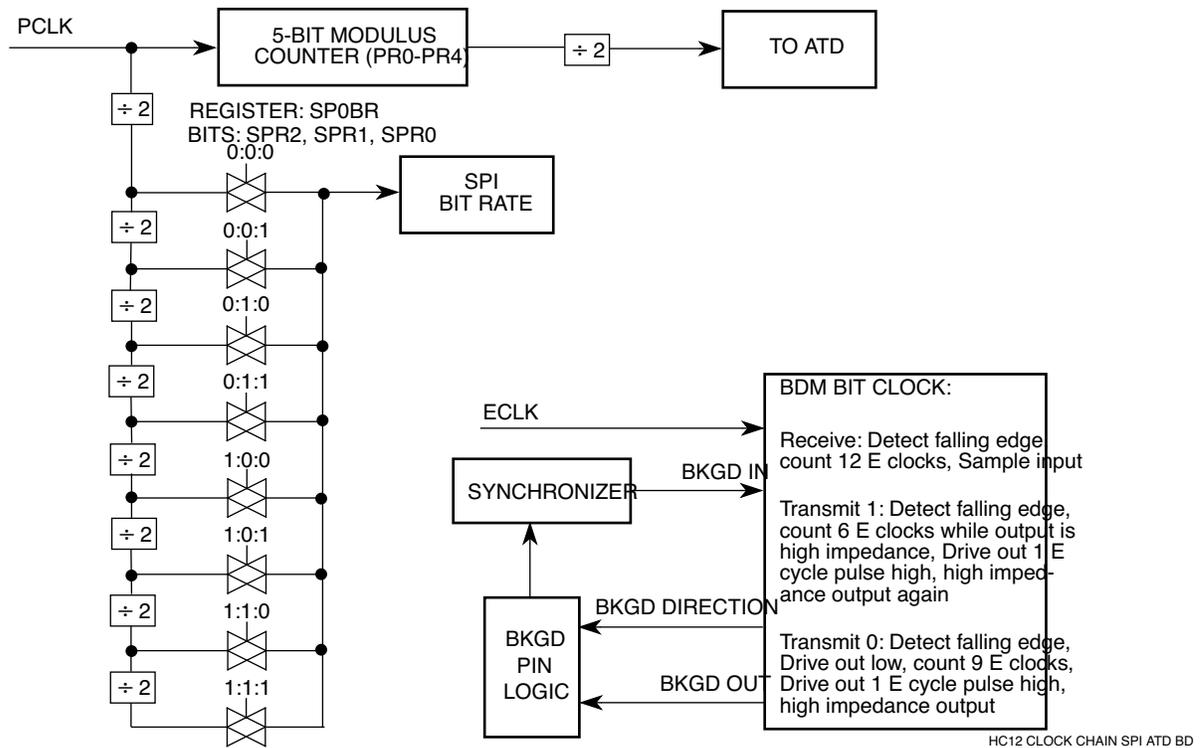
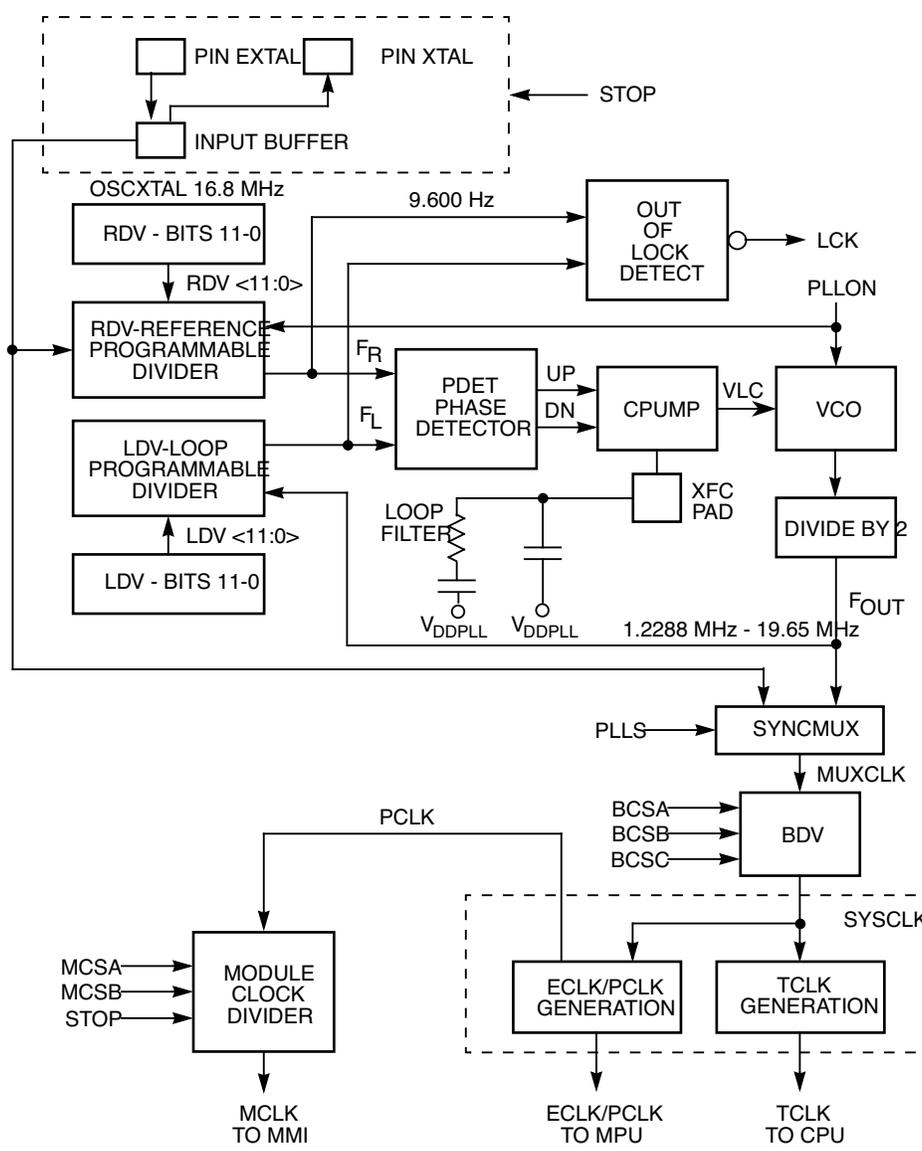


Figure 16 Clock Chain for SPI, ATD and BDM

## 12 Phase-Locked Loop

The phase-locked loop (PLL) allows slight adjustments in the frequency of the MC68HC812A4. The smallest increment of adjustment is  $\pm 9.6$  kHz to the output frequency ( $F_{out}$ ) rate assuming an input clock of 16.8 MHz (oscxtal) and a reference divider set to 1750. **Figure 17** shows the PLL dividers and a portion of the clock module.



**Figure 17 PLL Functional Diagram**

The PLL may be used to run the MCU from a different time base than the incoming crystal value. If the PLL is selected, it will continue to run when in wait or stop mode resulting in more power consumption than normal. To take full advantage of the reduced power consumption of stop mode, turn off the PLL before going into stop.

Although it is possible to set the divider to command a very high clock frequency, do not exceed the 16.8 MHz frequency limit for the MCU.

A passive external loop filter must be placed on the control line (XFC pad). The filter is a second-order, low-pass filter to eliminate the VCO input ripple. Values in the diagram are dependent upon the desired VCO operation.

## 12.1 PLL Register Description

### LDV — Loop Divider Registers

**\$0040, \$0041**

	Bit 15	14	13	12	11	10	9	Bit 8
	0	0	0	0	LDV11	LDV10	LDV9	LDV8
RESET:	0	0	0	0	1	1	1	1

	Bit 7	6	5	4	3	2	1	Bit 0
	LDV7	LDV6	LDV5	LDV4	LDV3	LDV2	LDV1	LDV0
RESET:	1	1	1	1	1	1	1	1

If the PLL is on, the count in the loop divider (LDV) 12-bit register effectively multiplies up from the PLL base frequency. Caution should be used not to exceed the maximum rated operating frequency for the CPU.

Read and write anytime.

### RDV — Reference Divider Registers

**\$0042, \$0043**

	Bit 15	14	13	12	11	10	9	Bit 8
	0	0	0	0	RDV11	RDV10	RDV9	RDV8
RESET:	0	0	0	0	1	1	1	1

	Bit 7	6	5	4	3	2	1	Bit 0
	RDV 7	RDV 6	RDV5	RDV4	RDV3	RDV2	RDV1	RDV0
RESET:	1	1	1	1	1	1	1	1

The count in the reference divider (RDV) 12-bit register effectively divides the crystal oscillator clock input.

Read and write anytime.

In the reset condition both LDV and RDV are set to the maximum count which produces an internal frequency at the phase detector of 8.2 kHz and a final output frequency of 16.8 MHz with a 16.8 MHz input clock.

### CLKCTL — Clock Control Register

**\$0047**

	Bit 7	6	5	4	3	2	1	Bit 0
	LCKF	PLLON	PLLS	BCSC	BCSB	BCSA	MCSB	MCSA
RESET:	0	0	0	0	0	0	0	0

Read and write anytime (LCKF is read only).

#### LCKF — Phase Lock Loop Circuit is Locked (Read Only)

0 = PLL is not on or not stable.

1 = After the phase lock loop circuit is turned on, indicates the PLL is at least half target frequency and no more than twice target frequency.

#### PLLON — Phase Lock Loop On

0 = Turns the PLL off.

1 = Turns on the phase lock loop circuit. Once the PLL is near the target frequency it will set the lock flag.

PLLS — Phase Lock Loop Select (PLL output or crystal input frequency)

0 = PLL is not selected, MUXCLK = crystal input frequency.

1 = After the phase lock loop circuit is locked, selects the PLL.

BCSA–BCSC — Base Clock Select

These bits determine the clock used by the main system including the CPU and buses. See **Table 21**. SYSCLK is the source clock for the MCU and is twice the bus rate. MUXCLK is either the PLL output or the crystal input frequency as selected by the PLLS bit.

**Table 21 Base Clock Select Bit Definition**

BCSC	BCSB	BCSA	SYSCLK Rate
0	0	0	SYSCLK = MUXCLK
0	0	1	Divide by 2
0	1	0	Divide by 4
0	1	1	Divide by 8
1	0	0	Divide by 16
1	0	1	Divide by 32
1	1	0	Divide by 64
1	1	1	Divide by 128

MCSA, MCSB — Module Clock Select

These bits determine the clock used by some sections of some of the modules such as the baud rate generators of the SCIs, the timer counter, the RTI and COP. See **Table 22**. MCLK is the module clock and PCLK is an internal bus rate clock.

**Table 22 Module Clock Select Bit Definition**

MCSB	MCSA	MCLK Rate
0	0	MCLK = PCLK
0	1	Divide by 2
1	0	Divide by 4
1	1	Divide by 8

The BCSx and MCSx bits can be changed with a single write access. In combination, these bits can be used to “throttle” the CPU clock rate without affecting the MCLK rate; timing and baud rates can remain constant as the processor speed is changed to match system requirements. This can save overall system power.

### 13 Standard Timer Module

The standard timer module consists of a 16-bit software-programmable counter driven by a prescaler. It contains eight complete 16-bit input capture/output compare channels and one 16-bit pulse accumulator.

This timer can be used for many purposes, including input waveform measurements while simultaneously generating an output waveform. Pulse widths can vary from less than a microsecond to many seconds. It can also generate PWM signals without CPU intervention.

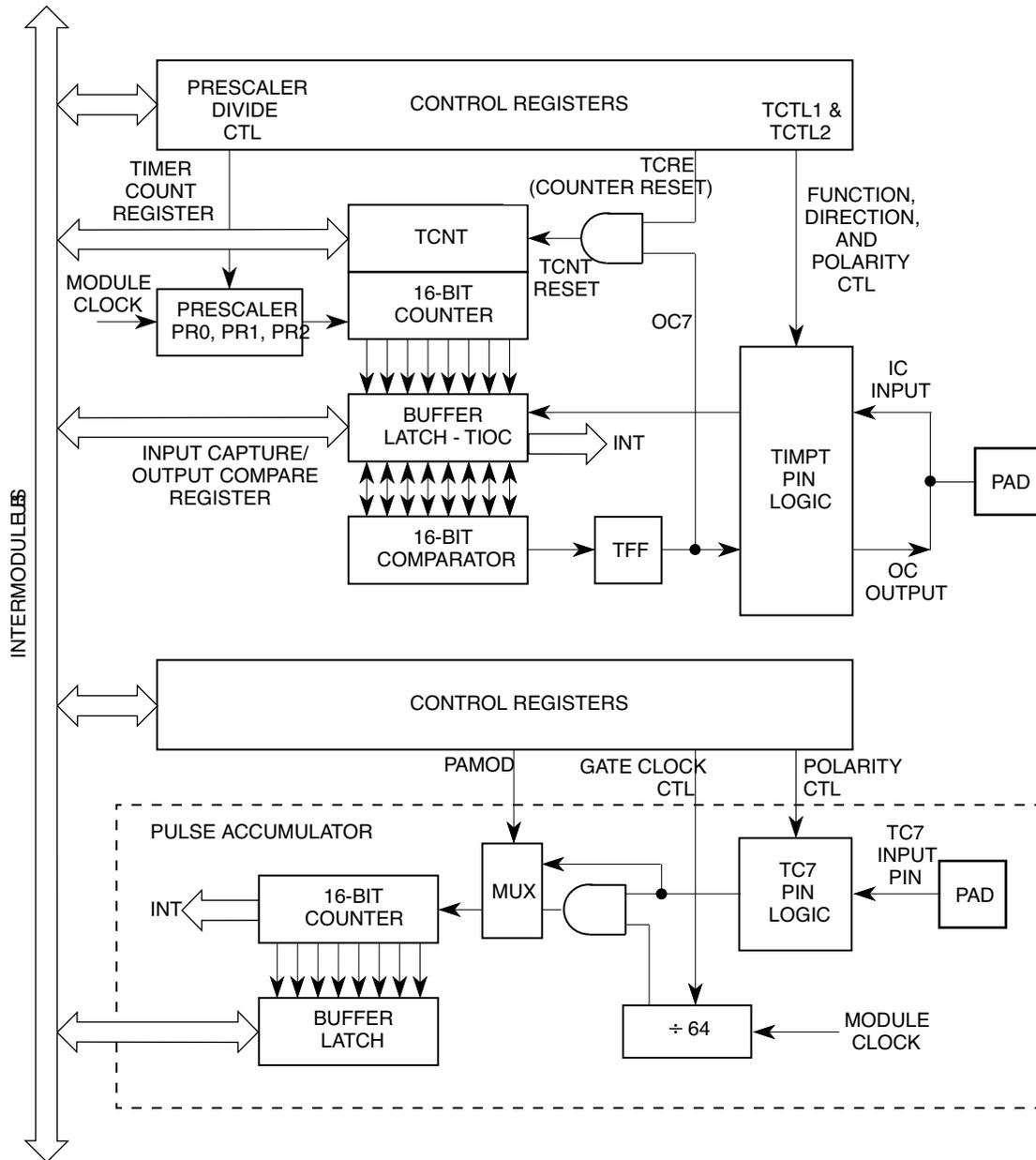


Figure 18 Timer Block Diagram — Input Capture, Output Compare, Pulse Accumulator

### 13.1 Timer Registers

Input/output pins default to general-purpose I/O lines until an internal function which uses that pin is specifically enabled. The timer overrides the state of the DDR to force the I/O state of each associated port line when an output compare using a port line is enabled. In these cases the data direction bits will have no affect on these lines.

When a pin is assigned to output an on-chip peripheral function, writing to this PORTTn bit does not affect the pin but the data is stored in an internal latch such that if the pin becomes available for general-purpose output the driven level will be the last value written to the PORTTn bit.

#### TIOS — Timer Input Capture/Output Compare Select

**\$0080**

	Bit 7	6	5	4	3	2	1	Bit 0
	IOS7	IOS6	IOS5	IOS4	IOS3	IOS2	IOS1	IOS0
RESET:	0	0	0	0	0	0	0	0

Read or write anytime.

#### IOS[7:0] — Input Capture or Output Compare Channel Designator

0 = The corresponding channel acts as an input capture

1 = The corresponding channel acts as an output compare.

#### CFORC — Timer Compare Force Register

**\$0081**

	Bit 7	6	5	4	3	2	1	Bit 0
	FOC7	FOC6	FOC5	FOC4	FOC3	FOC2	FOC1	FOC0
RESET:	0	0	0	0	0	0	0	0

Read anytime but will always return \$00 (1 state is transient). Write anytime.

#### FOC[7:0] — Force Output Compare Action for Channel 7-0

A write to this register with the corresponding data bit(s) set causes the action which is programmed for output compare "n" to occur immediately. The action taken is the same as if a successful comparison had just taken place with the TCn register except the interrupt flag does not get set.

#### OC7M — Output Compare 7 Mask Register

**\$0082**

	Bit 7	6	5	4	3	2	1	Bit 0
	OC7M7	OC7M6	OC7M5	OC7M4	OC7M3	OC7M2	OC7M1	OC7M0
RESET:	0	0	0	0	0	0	0	0

Read or write anytime.

The bits of OC7M correspond bit-for-bit with the bits of timer port (PORTT). Setting the OC7Mn will set the corresponding port to be an output port regardless of the state of the DDRTn bit when the corresponding TIOSn bit is set to be an output compare. This does not change the state of the DDRT bits.

#### OC7D — Output Compare 7 Data Register

**\$0083**

	Bit 7	6	5	4	3	2	1	Bit 0
	OC7D7	OC7D6	OC7D5	OC7D4	OC7D3	OC7D2	OC7D1	OC7D0
RESET:	0	0	0	0	0	0	0	0

Read or write anytime.

The bits of OC7D correspond bit-for-bit with the bits of timer port (PORTT). When a successful OC7 compare occurs, for each bit that is set in OC7M, the corresponding data bit in OC7D is stored to the corresponding bit of the timer port.

When the OC7Mn bit is set, a successful OC7 action will override a successful OC[6:0] compare action during the same cycle; therefore, the OCn action taken will depend on the corresponding OC7D bit.

**TCNT — Timer Count Register**

**\$0084–\$0085**

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

A full access for the counter register should take place in one clock cycle. A separate read/write for high byte and low byte will give a different result than accessing them as a word.

Read anytime.

Write has no meaning or effect in the normal mode; only writable in special modes (SMOD = 0).

The period of the first count after a write to the TCNT registers may be a different size because the write is not synchronized with the prescaler clock.

**TSCR — Timer System Control Register**

**\$0086**

	Bit 7	6	5	4	3	2	1	Bit 0
	TEN	TSWAI	TSBCK	TFFCA	0	0	0	0
RESET:	0	0	0	0	0	0	0	0

Read or write anytime.

**TEN — Timer Enable**

0 = Disables the timer, including the counter. Can be used for reducing power consumption.

1 = Allows the timer to function normally.

If for any reason the timer is not active, there is no  $\div 64$  clock for the pulse accumulator since the  $M\div 64$  is generated by the timer prescaler.

**TSWAI — Timer Stops While in Wait**

0 = Allows the timer to continue running during wait.

1 = Disables the timer when the MCU is in the wait mode. Timer interrupts cannot be used to get the MCU out of wait.

**TSBCK — Timer Stops While in Background Mode**

0 = Allows the timer to continue running while in background mode.

1 = Disables the timer whenever the MCU is in background mode. This is useful for emulation.

**TFFCA — Timer Fast Flag Clear All**

0 = Allows the timer flag clearing to function normally.

1 = For the TFLG1(\$8E), a read from an input capture or a write to the output compare channel (\$90–\$9F) causes the corresponding channel flag, CnF, to be cleared. For the TFLG2 (\$8F), any access to the TCNT register (\$84, \$85) would clear TOF flag. Any access to the PACNT register (\$A2, \$A3) would clear both PAOVF and PAIF flags in the PAFLG register (\$A1). This has the advantage of eliminating software overhead in a separate clear sequence. Extra care is required to avoid accidental flag clearing due to unintended accesses.

**TQCR** — Reserved

**\$0087**

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

**TCTL1** — Timer Control Register 1

**\$0088**

	Bit 7	6	5	4	3	2	1	Bit 0
	OM7	OL7	OM6	OL6	OM5	OL5	OM4	OL4
RESET:	0	0	0	0	0	0	0	0

**TCTL2** — Timer Control Register 2

**\$0089**

	Bit 7	6	5	4	3	2	1	Bit 0
	OM3	OL3	OM2	OL2	OM1	OL1	OM0	OL0
RESET:	0	0	0	0	0	0	0	0

Read or write anytime.

OMn — Output Mode

OLn — Output Level

These eight pairs of control bits are encoded to specify the output action to be taken as a result of a successful OCn compare. When either OMn or OLn is one, the pin associated with OCn becomes an output tied to OCn regardless of the state of the associated DDRT bit.

**Table 23 Compare Result Output Action**

OMn	OLn	Action
0	0	Timer disconnected from output pin logic
0	1	Toggle OCn output line
1	0	Clear OCn output line to zero
1	1	Set OCn output line to one

**TCTL3** — Timer Control Register 3

**\$008A**

	Bit 7	6	5	4	3	2	1	Bit 0
	EDG7B	EDG7A	EDG6B	EDG6A	EDG5B	EDG5A	EDG4B	EDG4A
RESET:	0	0	0	0	0	0	0	0

**TCTL4** — Timer Control Register 4

**\$008B**

	Bit 7	6	5	4	3	2	1	Bit 0
	EDG3B	EDG3A	EDG2B	EDG2A	EDG1B	EDG1A	EDG0B	EDG0A
RESET:	0	0	0	0	0	0	0	0

Read or write anytime.

EDGnB, EDGnA — Input Capture Edge Control

These eight pairs of control bits configure the input capture edge detector circuits.

**Table 24 Edge Detector Circuit Configuration**

EDGnB	EDGnA	Configuration
0	0	Capture disabled
0	1	Capture on rising edges only
1	0	Capture on falling edges only
1	1	Capture on any edge (rising or falling)

**TMSK1** — Timer Interrupt Mask 1

**\$008C**

	Bit 7	6	5	4	3	2	1	Bit 0
	C7I	C6I	C5I	C4I	C3I	C2I	C1I	C0I
RESET:	0	0	0	0	0	0	0	0

The bits in TMSK1 correspond bit-for-bit with the bits in the TFLG1 status register. If cleared, the corresponding flag is disabled from causing a hardware interrupt. If set, the corresponding flag is enabled to cause a hardware interrupt.

Read or write anytime.

C7I–C0I—Input Capture/Output Compare “x” Interrupt Enable.

**TMSK2** — Timer Interrupt Mask 2

**\$008D**

	Bit 7	6	5	4	3	2	1	Bit 0
	TOI	0	TPU	TDRB	TCRE	PR2	PR1	PR0
RESET:	0	0	1	1	0	0	0	0

Read or write anytime.

**TOI** — Timer Overflow Interrupt Enable

- 0 = Interrupt inhibited
- 1 = Hardware interrupt requested when TOF flag set

**TPU** — Timer Pull-Up Resistor Enable

This enable bit controls pull-up resistors on the timer port pins when the pins are configured as inputs.

- 1 = Enable pull-up resistor function
- 0 = Disable pull-up resistor function

**TDRB** — Timer Drive Reduction

This bit reduces the output driver size which can reduce power supply current and generated noise depending upon pin loading.

- 1 = Enable output drive reduction function
- 0 = Normal output drive capability

**TCRE** — Timer Counter Reset Enable

This bit allows the timer counter to be reset by a successful output compare 7 event.

- 0 = Counter reset inhibited and counter free runs
- 1 = Counter reset by a successful output compare 7

If TC7 = \$0000 and TCRE = 1, TCNT will stay at \$0000 continuously. If TC7 = \$FFFF and TCRE = 1, TOF will never get set even though TCNT will count from \$0000 through \$FFFF.

**PR2, PR1, PR0** — Timer Prescaler Select

These three bits specify the number of +2 stages that are to be inserted between the module clock (MCLK) and the timer free-running counter.

**Table 25 Prescaler Selection**

PR2	PR1	PR0	Prescale Factor
0	0	0	1
0	0	1	2
0	1	0	4
0	1	1	8
1	0	0	16
1	0	1	32
1	1	0	Reserved
1	1	1	Reserved

The newly selected prescale factor will not take effect until the next synchronized edge where all prescale counter stages equal zero.

**TFLG1 — Timer Interrupt Flag 1**

**\$008E**

	Bit 7	6	5	4	3	2	1	Bit 0
	C7F	C6F	C5F	C4F	C3F	C2F	C1F	C0F
RESET:	0	0	0	0	0	0	0	0

TFLG1 indicates when interrupt conditions have occurred. To clear a bit in the flag register, write a one to the bit.

Read anytime. Write used in the clearing mechanism (set bits cause corresponding bits to be cleared). Writing a zero will not effect current status of the bit.

When TFFCA bit in the TSCR register is set, a read from an input capture or a write into an output compare channel (\$90–\$9F) will cause the corresponding channel flag CnF to be cleared.

**C7F–C0F — Input Capture/Output Compare Channel “n” Flag.**

**TFLG2 — Timer Interrupt Flag 2**

**\$008F**

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

TFLG2 indicates when interrupt conditions have occurred. To clear a bit in the flag register, write a one to the bit.

Read anytime. Write used in clearing mechanism (bits set cause corresponding bits to be cleared).

Any access to TCNT will clear TFLG2 register if the TFFCA bit in TSCR register is set.

**TOF — Timer Overflow Flag**

Set when 16-bit free-running timer overflows from \$FFFF to \$0000. This bit is cleared automatically by a write to the TFLG2 register with bit 7 set. (See also TCRE control bit explanation.)

**TC0 — Timer Input Capture/Output Compare Register 0**

**\$0090–\$0091**

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

**TC1 — Timer Input Capture/Output Compare Register 1** **\$0092–\$0093**

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

**TC2 — Timer Input Capture/Output Compare Register 2** **\$0094–\$0095**

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

**TC3 — Timer Input Capture/Output Compare Register 3** **\$0096–\$0097**

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

**TC4 — Timer Input Capture/Output Compare Register 4** **\$0098–\$0099**

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

**TC5 — Timer Input Capture/Output Compare Register 5** **\$009A–\$009B**

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

**TC6 — Timer Input Capture/Output Compare Register 6** **\$009C–\$009D**

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

**TC7 — Timer Input Capture/Output Compare Register 7** **\$009E–\$009F**

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

Depending on the TIOS bit for the corresponding channel, these registers are used to latch the value of the free-running counter when a defined transition is sensed by the corresponding input capture edge detector or to trigger an output action for output compare.

Read anytime. Write anytime for output compare function. Writes to these registers have no meaning or effect during input capture. All timer input capture/output compare registers are reset to \$0000.

**PACTL** — Pulse Accumulator Control Register**\$00A0**

	Bit 7	6	5	4	3	2	1	Bit 0
	0	PAEN	PAMOD	PEDGE	CLK1	CLK0	PAOVI	PAI
RESET:	0	0	0	0	0	0	0	0

Read or write anytime.

**PAEN** — Pulse Accumulator System Enable

0 = Pulse Accumulator system disabled

1 = Pulse Accumulator system enabled

PAEN is independent from TEN.

**PAMOD** — Pulse Accumulator Mode

0 = Event counter mode

1 = Gated time accumulation mode

**PEDGE** — Pulse Accumulator Edge Control

For PAMOD = 0 (event counter mode)

0 = Falling edges on the pulse accumulator input pin (PT7/PAI) cause the count to be incremented

1 = Rising edges on the pulse accumulator input pin cause the count to be incremented

For PAMOD = 1 (gated time accumulation mode)

0 = Pulse accumulator input pin high enables M=64 clock to pulse accumulator and the trailing falling edge on the pulse accumulator input pin sets the PAIF flag.

1 = Pulse accumulator input pin low enables M=64 clock to pulse accumulator and the trailing rising edge on the pulse accumulator input pin sets the PAIF flag.

If the timer is not active, there is no ÷64 clock since the M=64 clock is generated by the timer prescaler.

**CLK1, CLK0** — Clock Select Register**Table 26 Clock Selection**

CLK1	CLK0	Selected Clock
0	0	Use timer prescaler clock as timer counter clock
0	1	Use PACLK as input to timer counter clock
1	0	Use PACLK/256 as timer counter clock frequency
1	1	Use PACLK/65536 as timer counter clock frequency

If the pulse accumulator is disabled (PAEN = 0), the prescaler clock from the timer is always used as an input clock to the timer counter. The change from one selected clock to the other happens immediately after these bits are written.

**PAOVI** — Pulse Accumulator Overflow Interrupt Enable

0 = Interrupt inhibited

1 = Interrupt requested if PAOVF is set

**PAI** — Pulse Accumulator Input Interrupt Enable

0 = Interrupt inhibited

1 = Interrupt requested if PAIF is set

**PAFLG** — Pulse Accumulator Flag Register**\$00A1**

	Bit 7	6	5	4	3	2	1	Bit 0
	0	0	0	0	0	0	PAOVF	PAIF
RESET:	0	0	0	0	0	0	0	0

Read or write anytime.

When TFFCA bit in the TSCR register is set, any access to the PACNT register will clear all the flags in the PAFLG register.

**PAOVF** — Pulse Accumulator Overflow Flag

Set when the 16-bit pulse accumulator overflows from \$FFFF to \$0000. This bit is cleared automatically by a write to the PAFLG register with bit 1 set.

**PAIF** — Pulse Accumulator Input Edge Flag

Set when the selected edge is detected at the pulse accumulator input pin. In event mode, the event edge triggers PAIF. In gated time accumulation mode, the trailing edge of the gate signal at the pulse accumulator input pin triggers PAIF. This bit is cleared automatically by a write to the PAFLG register with bit 0 set.

**PACNT** — 16-bit Pulse Accumulator Count Register**\$00A2–\$00A3**

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

Full count register access should take place in one clock cycle. A separate read/write for high byte and low byte will give a different result than accessing them as a word.

Read or write anytime.

**TIMTST** — Timer Test Register**\$00AD**

	Bit 7	6	5	4	3	2	1	Bit 0
	0	0	0	0	0	0	TCBYP	PCBYP
RESET:	0	0	0	0	0	0	0	0

Read anytime. Write only in special mode (SMODN = 0)

**TCBYP** — Timer Divider Chain Bypass

0 = Normal operation

1 = The 16-bit free-running timer counter is divided into two 8-bit halves and the prescaler is bypassed. The clock drives both halves directly.

**PCBYP** — Pulse Accumulator Divider Chain Bypass

0 = Normal operation

1 = The 16-bit pulse accumulator counter is divided into two 8-bit halves and the prescaler is bypassed. The clock drives both halves directly.

**PORTT** — Timer Port Data Register**\$00AE**

	Bit 7	6	5	4	3	2	1	Bit 0
	PT7	PT6	PT5	PT4	PT3	PT2	PT1	PT0
TIMER	I/OC7	I/OC6	I/OC5	I/OC4	I/OC3	I/OC2	I/OC1	I/OC0
PA	PAI							

Read anytime (inputs return pin level; outputs return pin driver input level). Write data stored in an internal latch (drives pins only if configured for output).

**NOTE**

Writes do not change pin state when the pin is configured for timer output. The minimum pulse width for pulse accumulator input should always be greater than two module clocks due to input synchronizer circuitry. The minimum pulse width for the input capture should always be greater than the width of two module clocks due to input synchronizer circuitry.

**DDRT** — Data Direction Register for Timer Port**\$00AF**

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

Read or write anytime.

0 = Configures the corresponding I/O pin for input only

1 = Configures the corresponding I/O pin for output

The timer forces the I/O state to be an output for each timer port pin associated with an enabled output compare. In these cases the data direction bits will not be changed but have no affect on the direction of these pins. The DDRT will revert to controlling the I/O direction of a pin when the associated timer output compare is disabled. Input captures do not override the DDRT settings.

**13.2 Timer Operation in Modes**

- STOP: Timer is off since both PCLK and ECLK are stopped.
- BDM: Timer keeps running, unless TSBCK = 1
- WAIT: Counters keep running, unless TSWAI = 1
- NORMAL: Timer keeps running, unless TEN = 0
- TEN = 0: All timer operations are stopped, registers may be accessed.  
Gated pulse accumulator +64 clock is also disabled.
- PAEN = 0: All pulse accumulator operations are stopped, registers may be accessed.

## 14 Multiple Serial Interface

The multiple serial interface (MSI) module consists of three independent serial I/O sub-systems: two serial communication interfaces (SCI0 and SCI1) and the serial peripheral interface (SPI0). Each serial pin shares function with the general-purpose port pins of port S. The SCI subsystems are NRZ-type systems that are compatible with standard RS-232 systems. These SCI systems have a new single wire operation mode which allows the unused pin to be available as general-purpose I/O. The SPI subsystem, which is compatible with the M68HC11 SPI, includes new features such as  $\overline{SS}$  output and bidirectional mode.

### 14.1 Block diagram

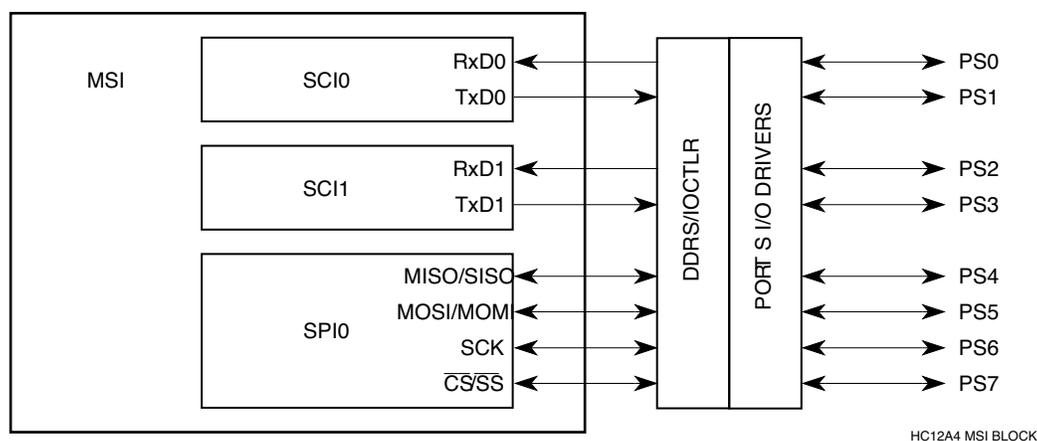


Figure 19 Multiple Serial Interface Block Diagram

### 14.2 Serial Communication Interface (SCI)

Two serial communication interfaces are available on the MC68HC812A4. These are NRZ format (one start, eight or nine data, and one stop bit) asynchronous communication systems with independent internal baud rate generation circuitry and SCI transmitters and receivers. They can be configured for eight or nine data bits (one of which may be designated as a parity bit, odd or even). If enabled, parity is generated in hardware for transmitted and received data. Receiver parity errors are flagged in hardware. The baud rate generator is based on a modulus counter, allowing flexibility in choosing baud rates. There is a receiver wake-up feature, an idle line detect feature, a loop-back mode, and various error detection features. Two port pins for each SCI provide the external interface for the transmitted data (TXD) and the received data (RXD).



### 14.2.2 SCI Baud Rate Generation

The basis of the SCI baud rate generator is a 13-bit modulus counter. This counter gives the generator the flexibility necessary to achieve a reasonable level of independence from the CPU operating frequency and still be able to produce standard baud rates with a minimal amount of error. The clock source for the generator comes from the M Clock.

**Table 27 Baud Rate Generation**

Desired SCI Baud Rate	BR Divisor for M = 4.0 MHz	BR Divisor for M = 8.0 MHz
110	2273	4545
300	833	1667
600	417	833
1200	208	417
2400	104	208
4800	52	104
9600	26	52
14400	17	35
19200	13	26
38400	—	13

### 14.2.3 Register Descriptions

Control and data registers for the SCI subsystem are described below. The memory address indicated for each register is the default address that is in use after reset. The entire 512-byte register block can be mapped to any 2-Kbyte boundary within the standard 64-Kbyte address space. Both SCI have identical control registers mapped in two blocks of eight bytes.

#### SC0BDH/SC1BDH — SCI Baud Rate Control Register

**\$00C0/\$00C8**

	Bit 7	6	5	4	3	2	1	Bit 0	
	BTST	BSPL	BRLD	SBR12	SBR11	SBR10	SBR9	SBR8	High
RESET:	0	0	0	0	0	0	0	0	

#### SC0BDL/SC1BDL — SCI Baud Rate Control Register

**\$00C1/\$00C9**

	Bit 7	6	5	4	3	2	1	Bit 0	
	SBR7	SBR6	SBR5	SBR4	SBR3	SBR2	SBR1	SBR0	Low
RESET:	0	0	0	0	0	1	0	0	

SCxBDH and SCxBDL are considered together as a 16-bit baud rate control register.

Read any time. Write SBR[12:0] anytime. Low order byte must be written for change to take effect. Write SBR[15:13] only in special modes. The value in SBR[12:0] determines the baud rate of the SCI. The desired baud rate is determined by the following formula:

$$\text{SCI Baud Rate} = \frac{\text{MCLK}}{16 \times \text{BR}}$$

which is equivalent to:

$$\text{BR} = \frac{\text{MCLK}}{16 \times \text{SCI Baud Rate}}$$

BR is the value written to bits SBR[12:0] to establish baud rate.

**NOTE**

The baud rate generator is disabled until TE or RE bit in SCxCR2 register is set for the first time after reset, and/or the baud rate generator is disabled when SBR[12:0] = 0.

BTST — Reserved for test function

BSPL — Reserved for test function

BRLD — Reserved for test function

**SC0CR1/SC1CR1 — SCI Control Register 1**

**\$00C2/\$00CA**

	Bit 7	6	5	4	3	2	1	Bit 0
	LOOPS	WOMS	RSRC	M	WAKE	ILT	PE	PT
RESET:	0	0	0	0	0	0	0	0

Read or write anytime.

**LOOPS — SCI LOOP Mode/Single Wire Mode Enable**

0 = SCI transmit and receive sections operate normally.

1 = SCI receive section is disconnected from the RXD pin and the RXD pin is available as general purpose I/O. The receiver input is determined by the RSRC bit. The transmitter output is controlled by the associated DDRS bit. Both the transmitter and the receiver must be enabled to use the LOOP or the single wire mode.

If the DDRS bit associated with the TXD pin is set during the LOOPS = 1, the TXD pin outputs the SCI waveform. If the DDRS bit associated with the TXD pin is clear during the LOOPS = 1, the TXD pin becomes high (IDLE line state) for RSRC = 0 and high impedance for RSRC = 1. Refer to **Table 28**.

**WOMS — Wired-OR Mode for Serial Pins**

This bit controls the two pins (TXD and RXD) associated with the SCIx section.

0 = Pins operate in a normal mode with both high and low drive capability. To affect the RXD bit, that bit would have to be configured as an output (via DDRS0/2) which is the single wire case when using the SCI. WOMS bit still affects general-purpose output on TXD and RXD pins when SCIx is not using these pins.

1 = Each pin operates in an open drain fashion if that pin is declared as an output.

**RSRC — Receiver Source**

When LOOPS = 1, the RSRC bit determines the internal feedback path for the receiver.

0 = Receiver input is connected to the transmitter internally (not TXD pin)

1 = Receiver input is connected to the TXD pin

**Table 28 Loop Mode Functions**

LOOPS	RSRC	DDRS1(3)	WOMS	Function of Port S Bit 1/3
0	x	x	x	Normal Operations
1	0	0	0/1	LOOP mode without TXD output (TXD = High Impedance)
1	0	1	0	LOOP mode with TXD output (CMOS)
1	0	1	1	LOOP mode with TXD output (open-drain)
1	1	0	x	Single wire mode without TXD output (the pin is used as receiver input only, TXD = High Impedance)
1	1	1	0	Single wire mode with TXD output (the output is also fed back to receiver input, CMOS)
1	1	1	1	Single wire mode for the receiving and transmitting (open-drain)

M — Mode (select character format)  
 0 = One start, eight data, one stop bit  
 1 = One start, eight data, ninth data, one stop bit

WAKE — Wakeup by Address Mark/Idle  
 0 = Wake up by IDLE line recognition  
 1 = Wake up by address mark (last data bit set)

ILT — Idle Line Type  
 Determines which of two types of idle line detection will be used by the SCI receiver.  
 0 = Short idle line mode is enabled.  
 1 = Long idle line mode is detected.

In the short mode, the SCI circuitry begins counting ones in the search for the idle line condition immediately after the start bit. This means that the stop bit and any bits that were ones before the stop bit could be counted in that string of ones, resulting in earlier recognition of an idle line.

In the long mode, the SCI circuitry does not begin counting ones in the search for the idle line condition until a stop bit is received. Therefore, the last byte's stop bit and preceding "1" bits do not affect how quickly an idle line condition can be detected.

PE — Parity Enable  
 0 = Parity is disabled.  
 1 = Parity is enabled.

PT — Parity Type  
 If parity is enabled, this bit determines even or odd parity for both the receiver and the transmitter.  
 0 = Even parity is selected. An even number of ones in the data character causes the parity bit to be zero and an odd number of ones causes the parity bit to be one.  
 1 = Odd parity is selected. An odd number of ones in the data character causes the parity bit to be zero and an even number of ones causes the parity bit to be one.

**SC0CR2/SC1CR2 — SCI Control Register 2** **\$00C3/\$00CB**

	Bit 7	6	5	4	3	2	1	Bit 0
	TIE	TCIE	RIE	ILIE	TE	RE	RWU	SBK
RESET:	0	0	0	0	0	0	0	0

Read or write anytime.

TIE — Transmit Interrupt Enable  
 0 = TDRE interrupts disabled  
 1 = SCI interrupt will be requested whenever the TDRE status flag is set.

TCIE — Transmit Complete Interrupt Enable  
 0 = TC interrupts disabled  
 1 = SCI interrupt will be requested whenever the TC status flag is set.

RIE — Receiver Interrupt Enable  
 0 = RDRF and OR interrupts disabled  
 1 = SCI interrupt will be requested whenever the RDRF status flag or the OR status flag is set.

ILIE — Idle Line Interrupt Enable  
 0 = IDLE interrupts disabled  
 1 = SCI interrupt will be requested whenever the IDLE status flag is set.

TE — Transmitter Enable

0 = Transmitter disabled

1 = SCI transmit logic is enabled and the TXD pin (port S bit 1/bit 3) is dedicated to the transmitter. The TE bit can be used to queue an idle preamble.

RE — Receiver Enable

0 = Receiver disabled

1 = Enables the SCI receive circuitry.

RWU — Receiver Wake-Up Control

0 = Normal SCI Receiver

1 = Enables the wake-up function and inhibits further receiver interrupts. Normally hardware wakes the receiver by automatically clearing this bit.

SBK — Send Break

0 = Break generator off

1 = Generate a break code (at least ten or eleven contiguous zeros).

As long as SBK remains set the transmitter will send zeros. When SBK is changed to zero, the current frame of all zeros is finished before the TxD line goes to the idle state. If SBK is toggled on and off, the transmitter will send only ten (or eleven) zeros and then revert to mark idle or sending data.

**SC0SR1/SC1SR1** — SCI Status Register 1

**\$00C4/\$00CC**

	Bit 7	6	5	4	3	2	1	Bit 0
	TDRE	TC	RDRF	IDLE	OR	NF	FE	PF
RESET:	1	1	0	0	0	0	0	0

The bits in these registers are set by various conditions in the SCI hardware and are automatically cleared by special acknowledge sequences. The receive related flag bits in SCxSR1 (RDRF, IDLE, OR, NF, FE, and PF) are all cleared by a read of the SCxSR1 register followed by a read of the transmit/receive data register low byte. However, only those bits which were set when SCxSR1 was read will be cleared by the subsequent read of the transmit/receive data register low byte. The transmit related bits in SCxSR1 (TDRE and TC) are cleared by a read of the SCxSR1 register followed by a write to the transmit/receive data register low byte.

Read anytime (used in auto clearing mechanism). Write has no meaning or effect.

TDRE — Transmit Data Register Empty Flag

New data will not be transmitted unless SCxSR1 is read before writing to the transmit data register. Reset sets this bit.

0 = SCxDR busy

1 = Any byte in the transmit data register is transferred to the serial shift register so new data may now be written to the transmit data register.

TC — Transmit Complete Flag

Flag is set when the transmitter is idle (no data, preamble, or break transmission in progress). Clear by reading SCxSR1 with TC set and then writing to SCxDR.

0 = Transmitter busy

1 = Transmitter is idle

RDRF — Receive Data Register Full Flag

Once cleared, IDLE is not set again until the RxD line has been active and becomes idle again. RDRF is set if a received character is ready to be read from SCxDR. Clear the RDRF flag by reading SCxSR1 with RDRF set and then reading SCxDR.

0 = SCxDR empty

1 = SCxDR full

**IDLE — Idle Line Detected Flag**

Receiver idle line is detected (the receipt of a minimum of 10/11 consecutive ones). This bit will not be set by the idle line condition when the RWU bit is set. Once cleared, IDLE will not be set again until after RDRF has been set (after the line has been active and becomes idle again).

- 0 = RxD line is idle
- 1 = RxD line is active

**OR — Overrun Error Flag**

New byte is ready to be transferred from the receive shift register to the receive data register and the receive data register is already full (RDRF bit is set). Data transfer is inhibited until this bit is cleared.

- 0 = No overrun
- 1 = Overrun detected

**NF — Noise Error Flag**

Set during the same cycle as the RDRF bit but not set in the case of an overrun (OR).

- 0 = Unanimous decision
- 1 = Noise on a valid start bit, any of the data bits, or on the stop bit

**FE — Framing Error Flag**

Set when a zero is detected where a stop bit was expected. Clear the FE flag by reading SCxSR1 with FE set and then reading SCxDR.

- 0 = Stop bit detected
- 1 = Zero detected rather than a stop bit

**PF — Parity Error Flag**

Indicates if received data's parity matches parity bit. This feature is active only when parity is enabled. The type of parity tested for is determined by the PT (parity type) bit in SCxCR1.

- 0 = Parity correct
- 1 = Incorrect parity detected

**SC0SR2/SC1SR2 — SCI Status Register 2****\$00C5/\$00CD**

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

Read anytime. Write has no meaning or effect.

**RAF — Receiver Active Flag**

This bit is controlled by the receiver front end. It is set during the RT1 time period of the start bit search. It is cleared when an idle state is detected or when the receiver circuitry detects a false start bit (generally due to noise or baud rate mismatch).

- 0 = A character is not being received
- 1 = A character is being received

**SC0DRH/SC1DRH — SCI Data Register High****\$00C6/\$00CE**

	Bit 7	6	5	4	3	2	1	Bit 0
RESET:	R8	T8	0	0	0	0	0	0

**SC0DRL/SC1DRL — SCI Data Register Low****\$00C7/\$00CF**

	Bit 7	6	5	4	3	2	1	Bit 0
RESET:	R7T7	R6T6	R5T5	R4T4	R3T3	R2T2	R1T1	R0T0

#### R8 — Receive Bit 8

Read anytime. Write has no meaning or affect.

This bit is the ninth serial data bit received when the SCI system is configured for nine-data-bit operation.

#### T8 — Transmit Bit 8

Read or write anytime.

This bit is the ninth serial data bit transmitted when the SCI system is configured for nine-data-bit operation. When using 9-bit data format this bit does not have to be written for each data word. The same value will be transmitted as the ninth bit until this bit is rewritten.

#### R7T7–R0T0 — Receive/Transmit Data Bits 7 to 0

Reads access the eight bits of the read-only SCI receive data register (RDR). Writes access the eight bits of the write-only SCI transmit data register (TDR). SCxDRL:SCxDRH form the 9-bit data word for the SCI. If the SCI is being used with a 7- or 8-bit data word, only SCxDRL needs to be accessed. If a 9-bit format is used, the upper register should be written first to ensure that it is transferred to the transmitter shift register with the lower register.

### 14.3 Serial Peripheral Interface (SPI)

The serial peripheral interface allows the MC68HC812A4 to communicate synchronously with peripheral devices and other microprocessors. The SPI system in the MC68HC812A4 can operate as a master or as a slave. The SPI is also capable of interprocessor communications in a multiple master system.

When the SPI is enabled, all pins that are defined by the configuration as inputs will be inputs regardless of the state of the DDRS bits for those pins. All pins that are defined as SPI outputs will be outputs only if the DDRS bits for those pins are set. Any SPI output whose corresponding DDRS bit is cleared can be used as a general-purpose input.

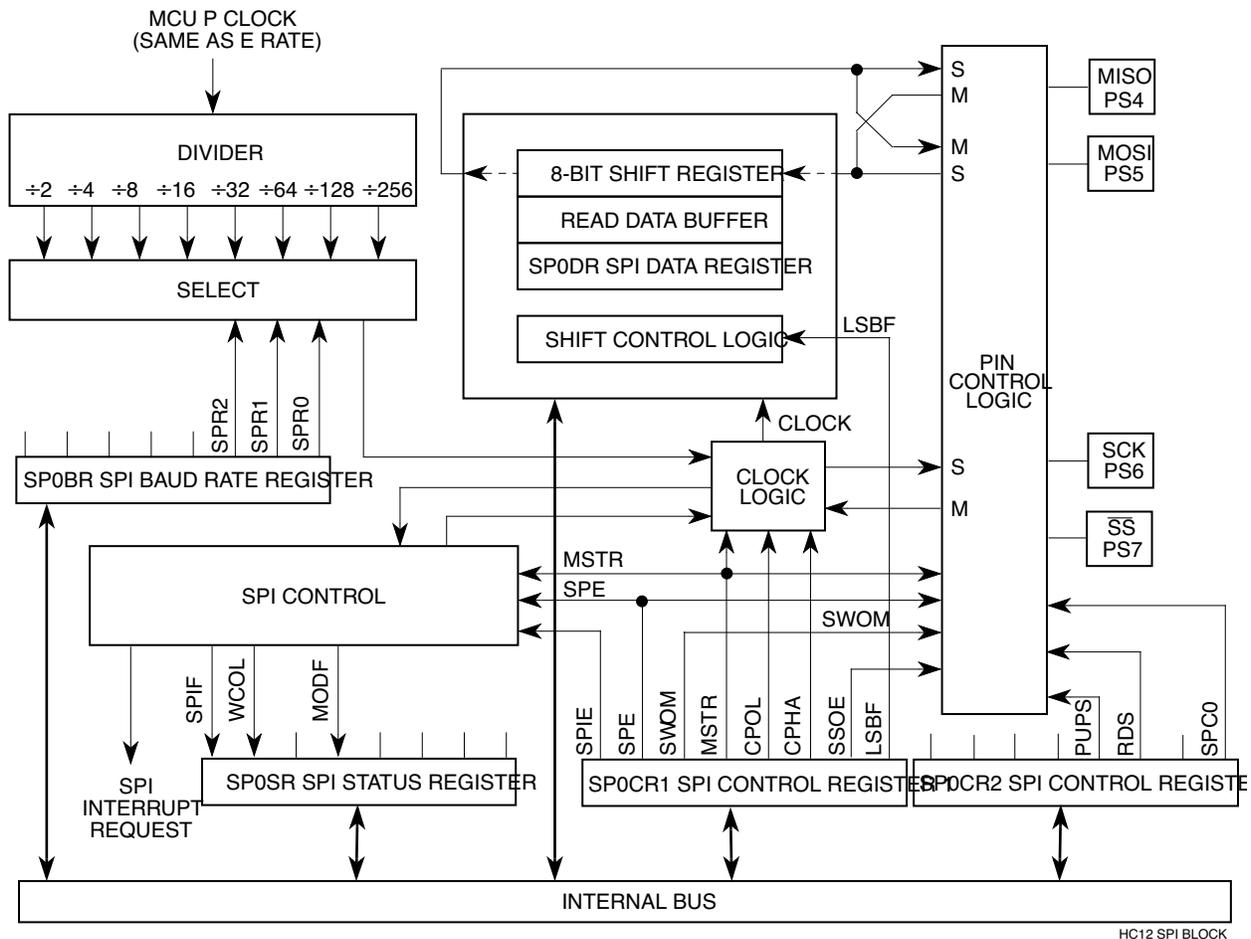
A bidirectional serial pin is possible using the DDRS as the direction control.

#### 14.3.1 SPI Baud Rate Generation

The E Clock is input to a divider series and the resulting SPI clock rate may be selected to be E divided by 2, 4, 8, 16, 32, 64, 128 or 256. Three bits in the SP0BR register control the SPI clock rate. This baud rate generator is activated only when SPI is in the master mode and serial transfer is taking place. Otherwise this divider is disabled to save power.

#### 14.3.2 SPI Operation

In the SPI system the 8-bit data register in the master and the 8-bit data register in the slave are linked to form a distributed 16-bit register. When a data transfer operation is performed, this 16-bit register is serially shifted eight bit positions by the SCK clock from the master so the data is effectively exchanged between the master and the slave. Data written to the SP0DR register of the master becomes the output data for the slave and data read from the SP0DR register of the master after a transfer operation is the input data from the slave.



**Figure 21 Serial Peripheral Interface Block Diagram**

A clock phase control bit (CPHA) and a clock polarity control bit (CPOL) in the SP0CR1 register select one of four possible clock formats to be used by the SPI system. The CPOL bit simply selects non-inverted or inverted clock. The CPHA bit is used to accommodate two fundamentally different protocols by shifting the clock by one half cycle or no phase shift.

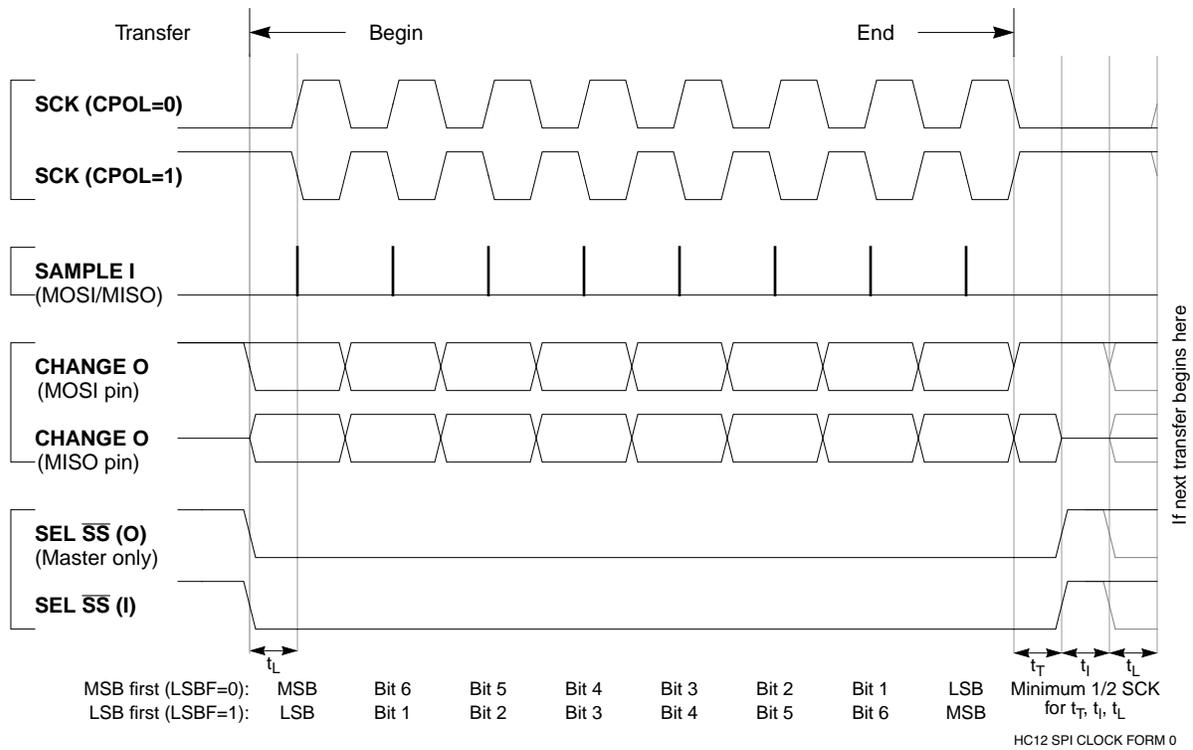


Figure 22 SPI Clock Format 0 (CPHA = 0)

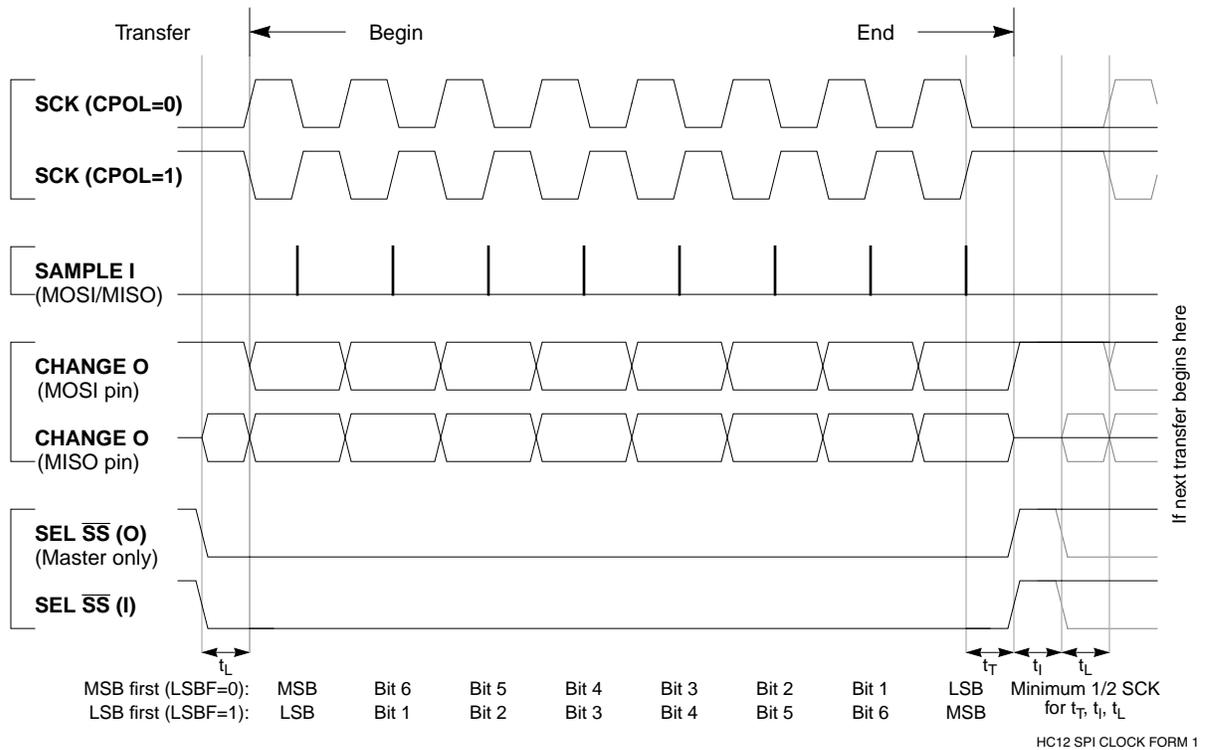


Figure 23 SPI Clock Format 1 (CPHA = 1)

### 14.3.3 $\overline{SS}$ Output

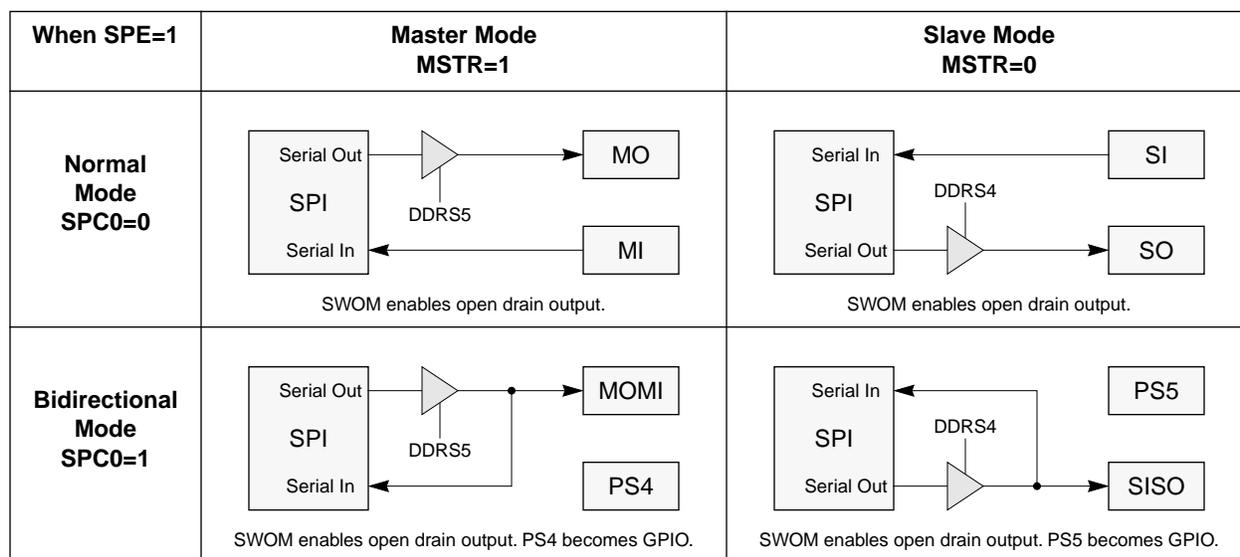
Available in master mode only,  $\overline{SS}$  output is enabled with the SSOE bit in the SP0CR1 register if the corresponding DDRS is set. The  $\overline{SS}$  output pin will be connected to the  $\overline{SS}$  input pin of the external slave device. The  $\overline{SS}$  output automatically goes low for each transmission to select the external device and it goes high during each idling state to deselect external devices.

**Table 29  $\overline{SS}$  Output Selection**

DDRS7	SSOE	Master Mode	Slave Mode
0	0	$\overline{SS}$ Input with MODF Feature	$\overline{SS}$ Input
0	1	Reserved	$\overline{SS}$ Input
1	0	General-Purpose Output	$\overline{SS}$ Input
1	1	$\overline{SS}$ Output	$\overline{SS}$ Input

### 14.3.4 Bidirectional Mode (MOMI or SISO)

In bidirectional mode, the SPI uses only one serial data pin for external device interface. The MSTR bit decides which pin to be used. The MOSI pin becomes serial data I/O (MOMI) pin for the master mode, and the MISO pin becomes serial data I/O (SISO) pin for the slave mode. The direction of each serial I/O pin depends on the corresponding DDRS bit.



**Figure 24 Normal Mode and Bidirectional Mode**

### 14.3.5 Register Descriptions

Control and data registers for the SPI subsystem are described below. The memory address indicated for each register is the default address that is in use after reset. The entire 512-byte register block can be mapped to any 2-Kbyte boundary within the standard 64-Kbyte address space. For more information refer to **6 Operating Modes and Resource Mapping**.

#### SP0CR1 — SPI Control Register 1

**\$00D0**

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

Read or write anytime.

**SPIE** — SPI Interrupt Enable

- 1 = Hardware interrupt sequence is requested each time the SPIF or MODF status flag is set
- 0 = SPI interrupts are inhibited

**SPE** — SPI System Enable

- 0 = SPI internal hardware is initialized and SPI system is in a low-power disabled state.
- 1 = PS[4:7] are dedicated to the SPI function

When MODF is set, SPE always reads zero. SP0CR1 must be written as part of a mode fault recovery sequence.

**SWOM** — Port S Wired-OR Mode

Controls not only SPI output pins but also the general-purpose output pins (PS[4:7]) which are not used by SPI.

- 0 = SPI and/or PS[4:7] output buffers operate normally
- 1 = SPI and/or PS[4:7] output buffers behave as open-drain outputs

**MSTR** — SPI Master/Slave Mode Select

- 0 = Slave mode
- 1 = Master mode

**CPOL, CPHA** — SPI Clock Polarity, Clock Phase

These two bits are used to specify the clock format to be used in SPI operations. When the clock polarity bit is cleared and data is not being transferred, the SCK pin of the master device is low. When CPOL is set, SCK idles high. See **Figure 22** and **Figure 23**.

**SSOE** — Slave Select Output Enable

The  $\overline{SS}$  output feature is enabled only in the master mode by asserting the SSOE and DDRS7.

**LSBF** — SPI LSB First enable

- 0 = Data is transferred most significant bit first
- 1 = Data is transferred least significant bit first

Normally data is transferred most significant bit first. This bit does not affect the position of the MSB and LSB in the data register. Reads and writes of the data register will always have MSB in bit 7.

**SP0CR2** — SPI Control Register 2

**\$00D1**

	Bit 7	6	5	4	3	2	1	Bit 0
	0	0	0	0	PUPS	RDS	0	SPC0
RESET:	0	0	0	0	1	0	0	0

Read or write anytime.

**PUPS** — Pull-Up Port S Enable

- 0 = No internal pull-ups on port S
- 1 = All port S input pins have an active pull-up device. If a pin is programmed as output, the pull-up device becomes inactive

**RDS** — Reduce Drive of Port S

- 0 = Port S output drivers operate normally
- 1 = All port S output pins have reduced drive capability for lower power and less noise

**SPC0** — Serial Pin Control 0

This bit decides serial pin configurations with MSTR control bit.

Pin Mode		SPC0 <sup>1</sup>	MSTR	MISO <sup>2</sup>	MOSI <sup>3</sup>	SCK <sup>4</sup>	$\overline{SS}$ <sup>5</sup>
#1	Normal	0	0	Slave Out	Slave In	SCK In	$\overline{SS}$ In
#2			1	Master In	Master Out	SCK Out	$\overline{SS}$ I/O
#3	Bidirectional	1	0	Slave I/O	GPI/O	SCK In	$\overline{SS}$ In
#4			1	GPI/O	Master I/O	SCK Out	$\overline{SS}$ I/O

1. The serial pin control 0 bit enables bidirectional configurations.
2. Slave output is enabled if  $DDRS4 = 1$ ,  $SS = 0$  and  $MSTR = 0$ . (#1, #3)
3. Master output is enabled if  $DDRS5 = 1$  and  $MSTR = 1$ . (#2, #4)
4. SCK output is enabled if  $DDRS6 = 1$  and  $MSTR = 1$ . (#2, #4)
5.  $\overline{SS}$  output is enabled if  $DDRS7 = 1$ ,  $SSOE = 1$  and  $MSTR = 1$ . (#2, #4)

### SP0BR — SPI Baud Rate Register

**\$00D2**

	Bit 7	6	5	4	3	2	1	Bit 0
	0	0	0	0	0	SPR2	SPR1	SPR0
RESET:	0	0	0	0	0	0	0	0

Read anytime. Write anytime.

At reset, E clock divided by 2 is selected.

### SPR[2:0] — SPI Clock (SCK) Rate Select Bits

These bits are used to specify the SPI clock rate.

**Table 30 SPI Clock Rate Selection**

SPR2	SPR1	SPR0	E Clock Divisor	Frequency at E Clock = 4 MHz	Frequency at E Clock = 8 MHz
0	0	0	2	2.0 MHz	4.0 MHz
0	0	1	4	1.0 MHz	2.0 MHz
0	1	0	8	500 kHz	1.0 MHz
0	1	1	16	250 kHz	500 kHz
1	0	0	32	125 kHz	250 kHz
1	0	1	64	62.5 kHz	125 kHz
1	1	0	128	31.3 kHz	62.5 kHz
1	1	1	256	15.6 kHz	31.3 kHz

### SP0SR — SPI Status Register

**\$00D3**

	Bit 7	6	5	4	3	2	1	Bit 0
	SPIF	WCOL	0	MODF	0	0	0	0
RESET:	0	0	0	0	0	0	0	0

Read anytime. Write has no meaning or effect.

### SPIF — SPI Interrupt Request

SPIF is set after the eighth SCK cycle in a data transfer and it is cleared by reading the SP0SR register (with SPIF set) followed by an access (read or write) to the SPI data register.

### WCOL — Write Collision Status Flag

The MCU write is disabled to avoid writing over the data being transferred. No interrupt is generated because the error status flag can be read upon completion of the transfer that was in progress at the time of the error. Automatically cleared by a read of the SP0SR (with WCOL set) followed by an access (read or write) to the SP0DR register.

0 = No write collision

1 = Indicates that a serial transfer was in progress when the MCU tried to write new data into the SP0DR data register.

### MODF — SPI Mode Error Interrupt Status Flag

This bit is set automatically by SPI hardware if the MSTR control bit is set and the slave select input pin becomes zero. This condition is not permitted in normal operation. In the case where DDRS bit 7 is set, the PS7 pin is a general-purpose output pin or  $\overline{SS}$  output pin rather than being dedicated as the  $\overline{SS}$  input for the SPI system. In this special case the mode fault function is inhibited and MODF remains cleared. This flag is automatically cleared by a read of the SP0SR (with MODF set) followed by a write to the SP0CR1 register.

### SP0DR — SPI Data Register

**\$00D5**

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

Read anytime (normally only after SPIF flag set). Write anytime (see WCOL write collision flag).

Reset does not affect this address.

This 8-bit register is both the input and output register for SPI data. Reads of this register are double buffered but writes cause data to be written directly into the serial shifter. In the SPI system the 8-bit data register in the master and the 8-bit data register in the slave are linked by the MOSI and MISO wires to form a distributed 16-bit register. When a data transfer operation is performed, this 16-bit register is serially shifted eight bit positions by the SCK clock from the master so the data is effectively exchanged between the master and the slave. Note that some slave devices are very simple and either accept data from the master without returning data to the master or pass data to the master without requiring data from the master.

## 14.4 Port S

In all modes, port S bits PS[7:0] can be used for either general-purpose I/O, or with the SCI and SPI subsystems. During reset, port S pins are configured as high-impedance inputs (DDRS is cleared).

### PORTS — Port S Data Register

**\$00D6**

	Bit 7	6	5	4	3	2	1	Bit 0
Pin	PS7	PS6	PS5	PS4	PS3	PS2	PS1	PS0
Function	$\overline{SS}$ CS	SCK	MOSI MOMI	MISO SISO	TXD1	RXD1	TXD0	RXD0

Read anytime (inputs return pin level; outputs return pin driver input level). Write data stored in internal latch (drives pins only if configured for output). Writes do not change pin state when pin configured for SPI or SCI output.

After reset all bits are configured as general-purpose inputs.

Port S shares function with the on-chip serial systems (SPI0 and SCI0/1).

**DDRS** — Data Direction Register for Port S**\$00D7**

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

Read or write anytime.

After reset, all general-purpose I/O are configured for input only.

0 = Configure the corresponding I/O pin for input only

1 = Configure the corresponding I/O pin for output

**DDRS2, DDRS0** — Data Direction for Port S Bit 2 and Bit 0

If the SCI receiver is configured for two-wire SCI operation, corresponding port S pins will be input regardless of the state of these bits.

**DDRS3, DDRS1** — Data Direction for Port S Bit 3 and Bit 1

If the SCI transmitter is configured for two-wire SCI operation, corresponding port S pins will be output regardless of the state of these bits.

**DDRS[6:4]** — Data Direction for Port S Bits 6 through 4

If the SPI is enabled and expects the corresponding port S pin to be an input, it will be an input regardless of the state of the DDRS bit. If the SPI is enabled and expects the bit to be an output, it will be an output ONLY if the DDRS bit is set.

**DDRS7** — Data Direction for Port S Bit 7

In SPI slave mode, DDRS7 has no meaning or effect; the PS7 pin is dedicated as the  $\overline{SS}$  input. In SPI master mode, DDRS7 determines whether PS7 is an error detect input to the SPI or a general-purpose or slave select output line.

## 15 Analog-to-Digital Converter

The ATD is an 8-channel, 8-bit, multiplexed-input successive-approximation analog-to-digital converter, accurate to  $\pm 1$  least significant bit (LSB). It does not require external sample and hold circuits because of the type of charge redistribution technique used. The ATD converter timing can be synchronized to the system P clock. The ATD module consists of a 16-word (32-byte) memory-mapped control register block used for control, testing and configuration.

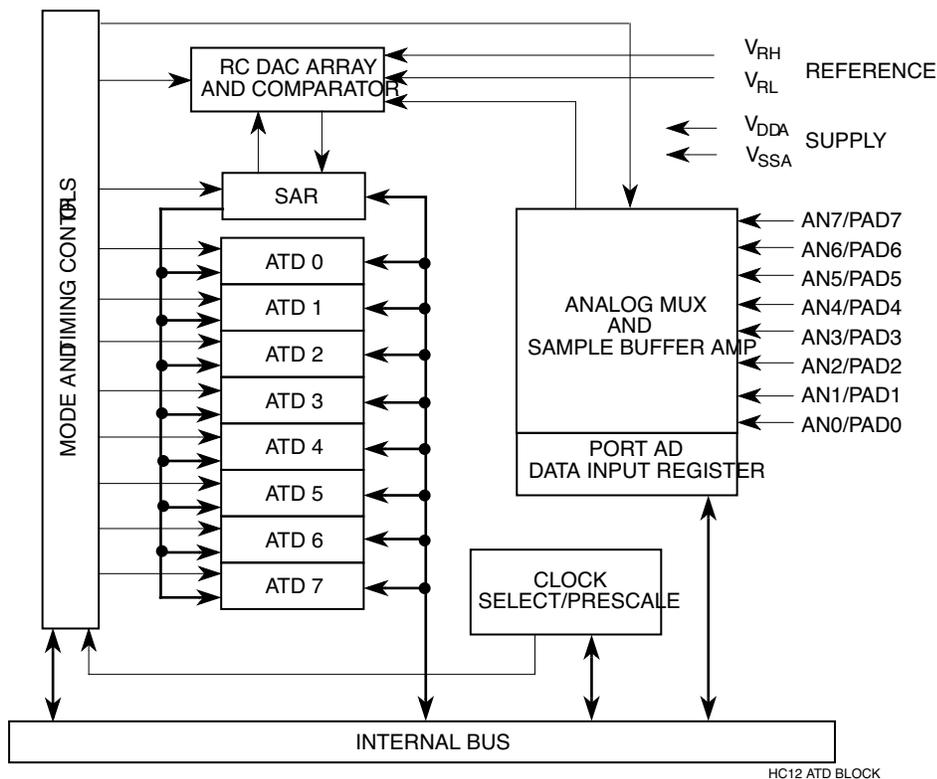


Figure 25 Analog-to-Digital Converter Block Diagram

### 15.1 Functional Description

A single conversion sequence consists of four or eight conversions, depending on the state of the select 8 channel mode (S8CM) bit when ATDCTL5 is written. There are eight basic conversion modes. In the non-scan modes, the SCF bit is set after the sequence of four or eight conversions has been performed and the ATD module halts. In the scan modes, the SCF bit is set after the first sequence of four or eight conversions has been performed, and the ATD module continues to restart the sequence. In both modes, the CCF bit associated with each register is set when that register is loaded with the appropriate conversion result. That flag is cleared automatically when that result register is read. The conversions are started by writing to the control registers.

## 15.2 ATD Registers

### ATDCTL0 — Reserved

\$0060

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

Writes to this register will abort the current conversion sequence.

### ATDCTL1 — Reserved

\$0061

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

Writes have no meaning or effect. Read in Special mode only.

### ATDCTL2 — ATD Control Register 2

\$0062

	Bit 7	6	5	4	3	2	1	Bit 0
	ADPU	AFFC	AWAI	0	0	0	ASCIE	ASCIF
RESET:	0	0	0	0	0	0	0	0

The ATD control register 2 and 3 are used to select the power up mode, interrupt control, and freeze control. Writes to these registers abort any current conversion sequence.

Read or write anytime except ASCIF bit, which cannot be written.

Bit positions ATDCTL2[4:2] and ATDCTL3[7:2] are unused and always read as zeros.

#### ADPU — ATD Power Up

0 = Disables the ATD, including the analog section for reduction in power consumption.

1 = Allows the ATD to function normally.

Software can disable the clock signal to the A/D converter and power down the analog circuits to reduce power consumption. When reset to zero, the ADPU bit aborts any conversion sequence in progress. Because the bias currents to the analog circuits are turned off, the ATD requires a period of recovery time to stabilize the analog circuits after setting the ADPU bit.

#### AFFC — ATD Fast Flag Clear All

0 = ATD flag clearing operates normally (read the status register before reading the result register to clear the associated CCF bit).

1 = Changes all ATD conversion complete flags to a fast clear sequence. Any access to a result register (ATD0–7) will cause the associated CCF flag to clear automatically if it was set at the time.

#### AWAI — ATD Wait Mode

0 = ATD continues to run when the MCU is in wait mode

1 = ATD stops to save power when the MCU is in wait mode

#### ASCIE — ATD Sequence Complete Interrupt Enable

0 = Disables ATD interrupt

1 = Enables ATD interrupt on sequence complete

#### ASCIF — ATD Sequence Complete Interrupt

Cannot be written in any mode.

0 = No ATD interrupt occurred

1 = ATD sequence complete

**ATDCTL3** — ATD Control Register 3

**\$0063**

	Bit 7	6	5	4	3	2	1	Bit 0
	0	0	0	0	0	0	FRZ1	FRZ0
RESET:	0	0	0	0	0	0	0	0

**FRZ1, FRZ0** — Background Debug (Freeze) Enable (suspend module operation at breakpoint)

When debugging an application, it is useful in many cases to have the ATD pause when a breakpoint is encountered. These two bits determine how the ATD will respond when background debug mode becomes active.

**Table 31 ATD Response to Background Debug Enable**

FRZ1	FRZ0	ATD Response
0	0	Continue conversions in active background mode
0	1	Reserved
1	0	Finish current conversion, then freeze
1	1	Freeze when BDM is active

**ATDCTL4** — ATD Control Register 4

**\$0064**

	Bit 7	6	5	4	3	2	1	Bit 0
	0	SMP1	SMP0	PRS4	PRS3	PRS2	PRS1	PRS0
RESET:	0	0	0	0	0	0	0	1

The ATD control register 4 is used to select the clock source and set up the prescaler. Writes to the ATD control registers initiate a new conversion sequence. If a write occurs while a conversion is in progress, the conversion is aborted and ATD activity halts until a write to ATDCTL5 occurs.

**SMP1, SMP0** — Select Sample Time

These bits are used to select one of four sample times after the buffered sample and transfer has occurred. Total conversion time depends on initial sample time (two ATD clocks), transfer time (four ATD clocks), final sample time (programmable, refer to **Table 32**), and resolution time (ten ATD clocks).

**Table 32 Final Sample Time Selection**

SMP1	SMP0	Final Sample Time	Total 8-Bit Conversion Time
0	0	2 ATD clock periods	18 ATD clock periods
0	1	4 ATD clock periods	20 ATD clock periods
1	0	8 ATD clock periods	24 ATD clock periods
1	1	16 ATD clock periods	32 ATD clock periods

**PRS4, PRS3, PRS2, PRS1, PRS0** — Select Divide-By Factor for ATD P-Clock Prescaler.

The binary value written to these bits selects the divide-by factor for the modulo counter-based prescaler. The P clock is divided by this value plus one and then fed into a ÷2 circuit to generate the ATD module clock. The divide-by-two circuit insures symmetry of the output clock signal. Clearing these bits causes the prescale value default to one which results in a ÷2 prescale factor. This signal is then fed into the ÷2 logic. The reset state divides the P clock by a total of four and is appropriate for nominal operation at 2 MHz. **Table 33** shows the divide-by operation and the appropriate range of system clock frequencies.

**Table 33 Clock Prescaler Values**

Prescale Value	Total Divisor	Max P Clock <sup>1</sup>	Min P Clock <sup>2</sup>
00000	÷2	4 MHz	1 MHz
00001	÷4	8 MHz	2 MHz
00010	÷6	8 MHz	3 MHz
00011	÷8	8 MHz	4 MHz
00100	÷10	8 MHz	5 MHz
00101	÷12	8 MHz	6 MHz
00110	÷14	8 MHz	7 MHz
00111	÷16	8 MHz	8 MHz
01xxx	Do Not Use		
1xxxx			

1. Maximum conversion frequency is 2 MHz. Maximum P clock divisor value will become maximum conversion rate that can be used on this ATD module.
2. Minimum conversion frequency is 500 kHz. Minimum P clock divisor value will become minimum conversion rate that this ATD can perform.

**ATDCTL5 — ATD Control Register 5**

**\$0065**

	Bit 7	6	5	4	3	2	1	Bit 0
	0	S8CM	SCAN	MULT	CD	CC	CB	CA
RESET:	0	0	0	0	0	0	0	0

The ATD control register 5 is used to select the conversion modes, the conversion channel(s), and initiate conversions.

Read or write any time. A write to ATDCTL5 initiates a new conversion sequence. If a conversion sequence is in progress when a write occurs, that sequence is aborted and the SCF and CCF bits are reset.

**S8CM — Select 8 Channel Mode**

- 0 = Conversion sequence consists of four conversions
- 1 = Conversion sequence consists of eight conversions

**SCAN — Enable Continuous Channel Scan**

- 0 = Single conversion sequence
- 1 = Continuous conversion sequences (scan mode)

When a conversion sequence is initiated by a write to the ATDCTL register, the user has a choice of performing a sequence of four (or eight, depending on the S8CM bit) conversions or continuously performing four (or eight) conversion sequences.

**MULT — Enable Multichannel Conversion**

- 0 = ATD sequencer runs all four or eight conversions on a **single** input channel selected via the CD, CC, CB, and CA bits.
- 1 = ATD sequencer runs each of the four or eight conversions on **sequential** channels in a specific group. Refer to **Table 34**.

**Table 34 Multichannel Mode Result Register Assignment**

S8CM	CD	CC	CB	CA	Channel Signal	Result in ADRx if MULT = 1	
0	0	0	0	0	AN0	ADR0	
			0	1	AN1	ADR1	
			1	0	AN2	ADR2	
			1	1	AN3	ADR3	
0	0	1	0	0	AN4	ADR0	
			0	1	AN5	ADR1	
			1	0	AN6	ADR2	
			1	1	AN7	ADR3	
0	1	0	0	0	Reserved	ADR0	
			0	1	Reserved	ADR1	
			1	0	Reserved	ADR2	
			1	1	Reserved	ADR3	
0	1	1	0	0	V <sub>RH</sub>	ADR0	
			0	1	V <sub>RL</sub>	ADR1	
			1	0	(V <sub>RH</sub> + V <sub>RL</sub> )/2	ADR2	
			1	1	TEST/Reserved	ADR3	
1	0	0	0	0	AN0	ADR0	
			0	0	1	AN1	ADR1
			0	1	0	AN2	ADR2
			0	1	1	AN3	ADR3
			1	0	0	AN4	ADR4
			1	0	1	AN5	ADR5
			1	1	0	AN6	ADR6
			1	1	1	AN7	ADR7
1	1	0	0	0	0	Reserved	ADR0
			0	0	1	Reserved	ADR1
			0	1	0	Reserved	ADR2
			0	1	1	Reserved	ADR3
			1	0	0	V <sub>RH</sub>	ADR4
			1	0	1	V <sub>RL</sub>	ADR5
			1	1	0	(V <sub>RH</sub> + V <sub>RL</sub> )/2	ADR6
			1	1	1	TEST/Reserved	ADR7

Shaded bits are “don't care” if MULT = 1 and the entire block of four or eight channels make up a conversion sequence. When MULT = 0, all four bits (CD, CC, CB, and CA) must be specified and a conversion sequence consists of four or eight consecutive conversions of the single specified channel.

**ATDSTAT** — ATD Status Register**\$0066**

	Bit 7	6	5	4	3	2	1	Bit 0
	SCF	0	0	0	0	CC2	CC1	CC0
RESET:	0	0	0	0	0	0	0	0

**ATDSTAT** — ATD Status Register**\$0067**

	Bit 7	6	5	4	3	2	1	Bit 0
	CCF7	CCF6	CCF5	CCF4	CCF3	CCF2	CCF1	CCF0
RESET:	0	0	0	0	0	0	0	0

The ATD status registers contain the flags indicating the completion of ATD conversions.

Normally, it is read-only. In special mode, the SCF bit and the CCF bits may also be written.

**SCF** — Sequence Complete Flag

This bit is set at the end of the conversion sequence when in the single conversion sequence mode (SCAN = 0 in ATDCTL5) and is set at the end of the first conversion sequence when in the continuous conversion mode (SCAN = 1 in ATDCTL5). When AFFC = 0, SCF is cleared when a write is performed to ATDCTL5 to initiate a new conversion sequence. When AFFC = 1, SCF is cleared after the first result register is read.

**CC[2:0]** — Conversion Counter for Current Sequence of Four or Eight Conversions

This 3-bit value reflects the contents of the conversion counter pointer in a four or eight count sequence. This value also reflects which result register will be written next, indicating which channel is currently being converted.

**CCF[7:0]** — Conversion Complete Flags

Each of these bits are associated with an individual ATD result register. For each register, this bit is set at the end of conversion for the associated ATD channel and remains set until that ATD result register is read. It is cleared at that time if AFFC bit is set, regardless of whether a status register read has been performed (i.e., a status register read is not a pre-qualifier for the clearing mechanism when AFFC = 1). Otherwise the status register must be read to clear the flag.

**ATDTEST** — ATD Test Register**\$0068**

	Bit 7	6	5	4	3	2	1	Bit 0
	SAR9	SAR8	SAR7	SAR6	SAR5	SAR4	SAR3	SAR2
RESET:	0	0	0	0	0	0	0	0

**ATDTEST** — ATD Test Register**\$0069**

	Bit 7	6	5	4	3	2	1	Bit 0
	SAR1	SAR0	RST	TSTOUT	TST3	TST2	TST1	TST0
RESET:	0	0	0	0	0	0	0	0

The test registers control various special modes which are used during manufacturing. The test register can be read or written only in the special modes. In the normal modes, reads of the test register return zero and writes have no effect.

**SAR[9:0]** — SAR Data

Reads of this byte return the current value in the SAR. Writes to this byte change the SAR to the value written. Bits SAR[9:2] reflect the eight SAR bits used during the resolution process for an 8-bit result. SAR1 and SAR0 are reserved to allow future derivatives to increase ATD resolution to 10 bits.

**RST** — Module Reset Bit

When set, this bit causes all registers and activity in the module to assume the same state as out of power-on reset (except for ADPU bit in ATDCTL2, which remains set, allowing the ATD module to remain enabled).

**TSTOUT** — Multiplex Output of TST[3:0] (Factory Use)

**TST[3:0]** — Test Bits 3 to 0 (Reserved)

Selects one of 16 reserved factory testing modes

**PORTAD** — Port AD Data Input Register

**\$006F**

	Bit 7	6	5	4	3	2	1	Bit 0
	PAD7	PAD6	PAD5	PAD4	PAD3	PAD2	PAD1	PAD0
RESET:	0	0	0	0	0	0	0	0

**PAD[7:0]** — Port AD Data Input Bits

After reset these bits reflect the state of the input pins.

May be used for general-purpose digital input. When the software reads PORTAD, it obtains the digital levels that appear on the corresponding port AD pins. Pins with signals not meeting  $V_{IL}$  or  $V_{IH}$  specifications will have an indeterminate value. Writes to this register have no meaning at any time.

**ADR0H** — A/D Converter Result Register 0

**\$0070**

**ADR1H** — A/D Converter Result Register 1

**\$0072**

**ADR2H** — A/D Converter Result Register 2

**\$0074**

**ADR3H** — A/D Converter Result Register 3

**\$0076**

**ADR4H** — A/D Converter Result Register 4

**\$0078**

**ADR5H** — A/D Converter Result Register 5

**\$007A**

**ADR6H** — A/D Converter Result Register 6

**\$007C**

**ADR7H** — A/D Converter Result Register 7

**\$007E**

\$007x	Bit 7	6	5	4	3	2	1	Bit 0
	Bit 7	6	5	4	3	2	1	Bit 0
RESET:	0	0	0	0	0	0	0	0

**ADRxH[7:0]** — ATD Conversion result

The reset condition for these registers is undefined.

These bits contain the left justified, unsigned result from the ATD conversion. The channel from which this result was obtained is dependent on the conversion mode selected. These registers are always read-only in normal mode.

**15.3 ATD Mode Operation**

**STOP** — Causes all clocks to halt (if the S bit in the CCR is zero). The system is placed in a minimum-power standby mode. This aborts any conversion sequence in progress.

**WAIT** — ATD conversion continues unless AWAI bit in ATDCTL2 register is set.

**BDM** — Debug options available as set in register ATDCTL3.

**USER** — ATD continues running unless ADPU is cleared.

**ADPU** — ATD operations are stopped if ADPU = 0, but registers are accessible.

## 16 Development Support

Development support involves complex interactions between MC68HC812A4 resources and external development systems. The following section concerns instruction queue and queue tracking signals, background debug mode, and instruction tagging.

### 16.1 Instruction Queue

The CPU12 instruction queue provides at least three bytes of program information to the CPU when instruction execution begins. The CPU12 always completely finishes executing an instruction before beginning to execute the next instruction. Status signals IPIPE[1:0] provide information about data movement in the queue and indicate when the CPU begins to execute instructions. This makes it possible to monitor CPU activity on a cycle-by-cycle basis for debugging. Information available on the IPIPE[1:0] pins is time multiplexed. External circuitry can latch data movement information on rising edges of the E-clock signal; execution start information can be latched on falling edges. **Table 35** shows the meaning of data on the pins.

**Table 35 IPIPE Decoding**

Data Movement — IPIPE[1:0] Captured at Rising Edge of E Clock <sup>1</sup>		
IPIPE[1:0]	Mnemonic	Meaning
0:0	—	No Movement
0:1	LAT	Latch Data From Bus
1:0	ALD	Advance Queue and Load From Bus
1:1	ALL	Advance Queue and Load From Latch
Execution Start — IPIPE[1:0] Captured at Falling Edge of E Clock <sup>2</sup>		
IPIPE[1:0]	Mnemonic	Meaning
0:0	—	No Start
0:1	INT	Start Interrupt Sequence
1:0	SEV	Start Even Instruction
1:1	SOD	Start Odd Instruction

1. Refers to data that was on the bus at the previous E falling edge.

2. Refers to bus cycle starting at this E falling edge.

Program information is fetched a few cycles before it is used by the CPU. In order to monitor cycle-by-cycle CPU activity, it is necessary to externally reconstruct what is happening in the instruction queue. Internally the MCU only needs to buffer the data from program fetches. For system debug it is necessary to keep the data and its associated address in the reconstructed instruction queue. The raw signals required for reconstruction of the queue are ADDR, DATA, R/W, ECLK, and status signals IPIPE[1:0].

The instruction queue consists of two 16-bit queue stages and a holding latch on the input of the first stage. To advance the queue means to move the word in the first stage to the second stage and move the word from either the holding latch or the data bus input buffer into the first stage. To start even (or odd) instruction means to execute the opcode in the high-order (or low-order) byte of the second stage of the instruction queue.

### 16.2 Background Debug Mode

Background debug mode (BDM) is used for system development, in-circuit testing, field testing, and programming. BDM is implemented in on-chip hardware and provides a full set of debug options.

Because BDM control logic does not reside in the CPU, BDM hardware commands can be executed while the CPU is operating normally. The control logic generally uses CPU dead cycles to execute these

commands, but can steal cycles from the CPU when necessary. Other BDM commands are firmware based, and require the CPU to be in active background mode for execution. While BDM is active, the CPU executes a firmware program located in a small on-chip ROM that is available in the standard 64-Kbyte memory map only while BDM is active.

The BDM control logic communicates with an external host development system serially, via the BKGD pin. This single-wire approach minimizes the number of pins needed for development support.

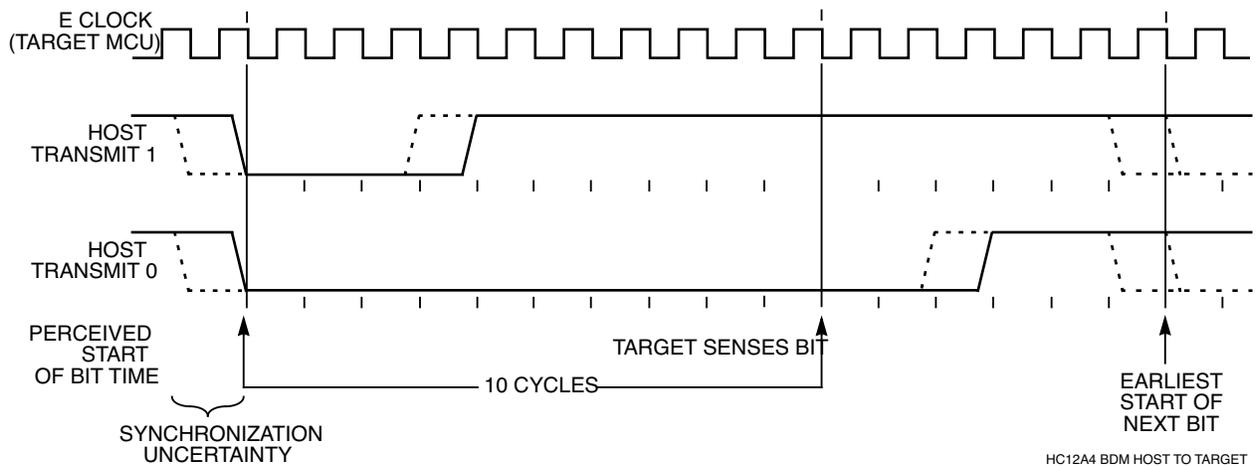
### 16.2.1 BDM Serial Interface

The BDM serial interface requires the external controller to generate a falling edge on the BKGD pin to indicate the start of each bit time. The external controller provides this falling edge whether data is transmitted or received.

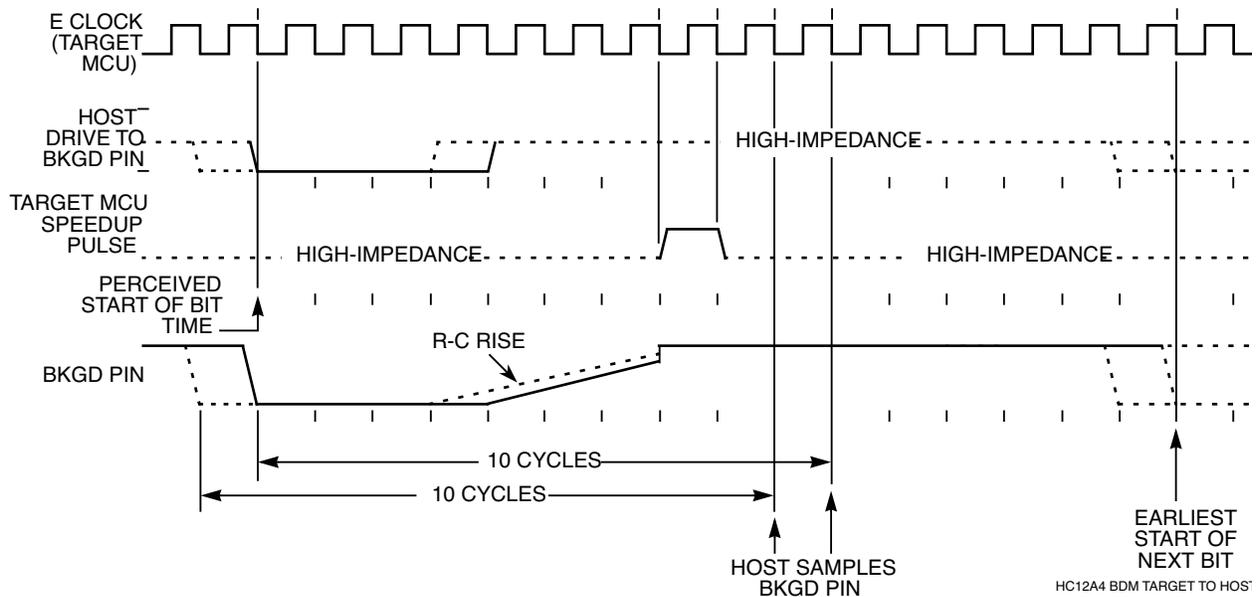
BKGD is a pseudo-open-drain pin that can be driven either by an external controller or by the MCU. Data is transferred MSB first at 16 E-clock cycles per bit (nominal speed). The interface times out if 512 E-clock cycles occur between falling edges from the host. The hardware clears the command register when this time-out occurs.

The BKGD pin can receive a high or low level or transmit a high or low level. The following diagrams show timing for each of these cases. Interface timing is synchronous to MCU clocks but asynchronous to the external host. The internal clock signal is shown for reference in counting cycles.

**Figure 26** shows an external host transmitting a logic one or zero to the BKGD pin of a target M68HC12 MCU. The host is asynchronous to the target so there is a 0-to-1 cycle delay from the host-generated falling edge to where the target perceives the beginning of the bit time. Ten target E cycles later, the target senses the bit level on the BKGD pin. Typically the host actively drives the pseudo-open-drain BKGD pin during host-to-target transmissions to speed up rising edges. Since the target does not drive the BKGD pin during this period, there is no need to treat the line as an open-drain signal during host-to-target transmissions.

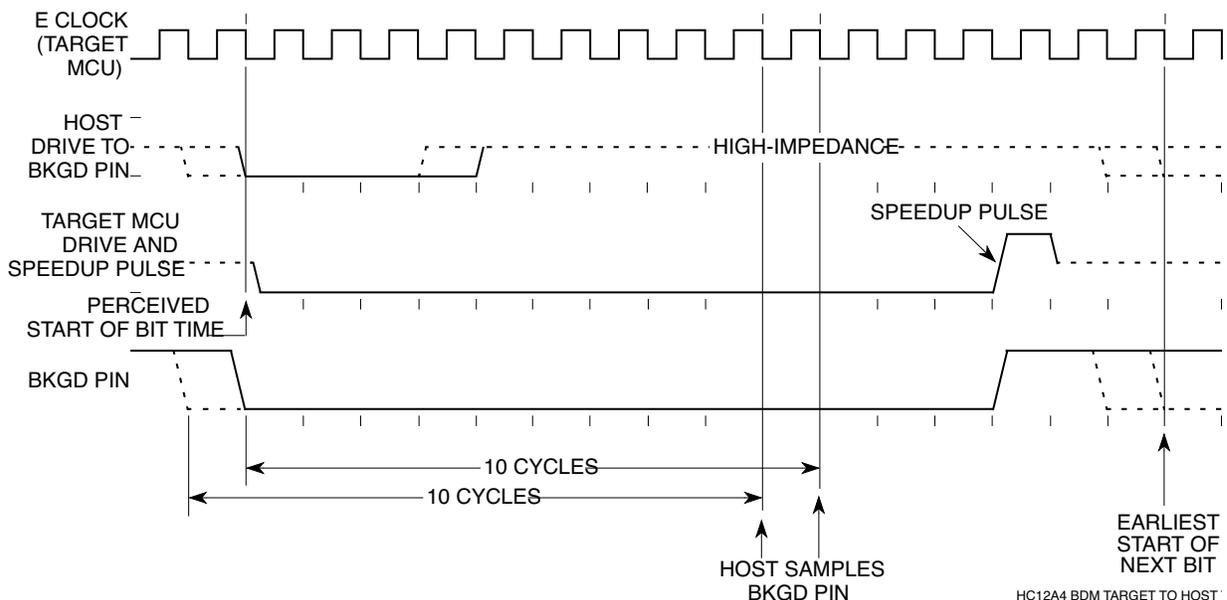


**Figure 26 BDM Host to Target Serial Bit Timing**



**Figure 27 BDM Target to Host Serial Bit Timing (Logic 1)**

**Figure 27** shows the host receiving a logic one from the target MC68HC812A4 MCU. Since the host is asynchronous to the target MCU, there is a 0-to-1 cycle delay from the host-generated falling edge on BKGD to the perceived start of the bit time in the target MCU. The host holds the BKGD pin low long enough for the target to recognize it (at least two target E cycles). The host must release the low drive before the target MCU drives a brief active-high speed-up pulse seven cycles after the perceived start of the bit time. The host should sample the bit level about ten cycles after it started the bit time.



**Figure 28 BDM Target to Host Serial Bit Timing (Logic 0)**

**Figure 28** shows the host receiving a logic zero from the target MC68HC812A4 MCU. Since the host is asynchronous to the target MCU, there is a 0-to-1 cycle delay from the host-generated falling edge on BKGD to the perceived start of the bit time in the target MCU. The host holds the BKGD pin low long enough for the target to recognize it (at least two target E cycles). The host must release the low drive before the target MCU drives a brief active-low speed-up pulse seven cycles after the perceived start of the bit time. The host should sample the bit level about ten cycles after it started the bit time.

on BKGD to the start of the bit time as perceived by the target MCU. The host initiates the bit time but the target MC68HC812A4 finishes it. Since the target wants the host to receive a logic zero, it drives the BKGD pin low for 13 E-clock cycles, then briefly drives it high to speed up the rising edge. The host samples the bit level about ten cycles after starting the bit time.

### 16.2.2 Enabling BDM Firmware Commands

BDM is available in all operating modes, but must be made active before firmware commands can be executed. BDM is enabled by setting the ENBDM bit in the BDM STATUS register via the single wire interface (using a hardware command; WRITE\_BD\_BYTE at \$FF01). BDM must then be activated to map BDM registers and ROM to addresses \$FF00 to \$FFFF and to put the MCU in active background mode.

After the firmware is enabled, BDM can be activated by the hardware BACKGROUND command, by the BDM tagging mechanism, or by the CPU BGND instruction. An attempt to activate BDM before firmware has been enabled causes the MCU to resume normal instruction execution after a brief delay.

BDM becomes active at the next instruction boundary following execution of the BDM BACKGROUND command, but tags activate BDM before a tagged instruction is executed.

In special single-chip mode, background operation is enabled and active immediately out of reset. This active case replaces the M68HC11 boot function, and allows programming a system with blank memory.

While BDM is active, a set of BDM control registers are mapped to addresses \$FF00 to \$FF06. The BDM control logic uses these registers which can be read anytime by BDM logic, not user programs. Refer to **16.2.4 BDM Registers** for detailed descriptions.

Some on-chip peripherals have a BDM control bit which allows suspending the peripheral function during BDM. For example, if the timer control is enabled, the timer counter is stopped while in BDM. Once normal program flow is continued, the timer counter is re-enabled to simulate real-time operations.

### 16.2.3 BDM Commands

All BDM command opcodes are eight bits long, and can be followed by an address and/or data, as indicated by the instruction. These commands do not require the CPU to be in active BDM mode for execution.

The host controller must wait 150 cycles for a non-intrusive BDM command to execute before another command can be sent. This delay includes 128 cycles for the maximum delay for a dead cycle. For data read commands, the host must insert this delay between sending the address and attempting to read the data.

BDM logic retains control of the internal buses until a read or write is completed. If an operation can be completed in a single cycle, it does not intrude on normal CPU operation. However, if an operation requires multiple cycles, CPU clocks are frozen until the operation is complete.

**Table 36 BDM Commands Implemented in Hardware**

Command	Opcode (Hex)	Data	Description
BACKGROUND	90	None	Enter background mode (if firmware enabled).
READ_BD_BYTE	E4	16-bit address 16-bit data out	Read from memory with BDM in map (may steal cycles if external access) data for odd address on low byte, data for even address on high byte.
STATUS <sup>1</sup>	E4	FF01, 0000 0000 (out)	READ_BD_BYTE \$FF01. Running user code (BGND instruction is not allowed).
		FF01, 1000 0000 (out)	READ_BD_BYTE \$FF01. BGND instruction is allowed.
		FF01, 1100 0000 (out)	READ_BD_BYTE \$FF01. Background mode active (waiting for single wire serial command).
READ_BD_WORD	EC	16-bit address 16-bit data out	Read from memory with BDM in map (may steal cycles if external access) must be aligned access.
READ_BYTE	E0	16-bit address 16-bit data out	Read from memory with BDM out of map (may steal cycles if external access) data for odd address on low byte, data for even address on high byte.
READ_WORD	E8	16-bit address 16-bit data out	Read from memory with BDM out of map (may steal cycles if external access) must be aligned access.
WRITE_BD_BYTE	C4	16-bit address 16-bit data in	Write to memory with BDM in map (may steal cycles if external access) data for odd address on low byte, data for even address on high byte.
ENABLE_FIRMWARE <sup>2</sup>	C4	FF01, 1xxx xxxx(in)	Write byte \$FF01, set the ENBDM bit. This allows execution of commands which are implemented in firmware. Typically, read STATUS, OR in the MSB, write the result back to STATUS.
WRITE_BD_WORD	CC	16-bit address 16-bit data in	Write to memory with BDM in map (may steal cycles if external access) must be aligned access.
WRITE_BYTE	C0	16-bit address 16-bit data in	Write to memory with BDM out of map (may steal cycles if external access) data for odd address on low byte, data for even address on high byte.
WRITE_WORD	C8	16-bit address 16-bit data in	Write to memory with BDM out of map (may steal cycles if external access) must be aligned access.

1. STATUS command is a specific case of the READ\_BD\_BYTE command.

2. ENABLE\_FIRMWARE is a specific case of the WRITE\_BD\_BYTE command.

The CPU must be in background mode to execute commands that are implemented in the BDM ROM. The BDM ROM is located at \$FF20 to \$FFFF while BDM is active. There are also seven bytes of BDM registers which are located at \$FF00 to \$FF06 while BDM is active. The CPU executes code from this ROM to perform the requested operation. These commands are shown in **Table 37**.

**Table 37 BDM Firmware Commands**

Command	Opcode (Hex)	Data	Description
READ_NEXT	62	16-bit data out	X = X + 2; Read next word pointed-to by X
READ_PC	63	16-bit data out	Read program counter
READ_D	64	16-bit data out	Read D accumulator
READ_X	65	16-bit data out	Read X index register
READ_Y	66	16-bit data out	Read Y index register
READ_SP	67	16-bit data out	Read stack pointer
WRITE_NEXT	42	16-bit data in	X = X + 2; Write next word pointed-to by X
WRITE_PC	43	16-bit data in	Write program counter
WRITE_D	44	16-bit data in	Write D accumulator
WRITE_X	45	16-bit data in	Write X index register
WRITE_Y	46	16-bit data in	Write Y index register
WRITE_SP	47	16-bit data in	Write stack pointer
GO	08	None	Go to user program
TRACE1	10	None	Execute one user instruction then return to BDM
TAGGO	18	None	Enable tagging and go to user program

### 16.2.4 BDM Registers

Seven BDM registers are mapped into the standard 64-Kbyte address space when BDM is active. The registers can be accessed with the hardware READ\_BD and WRITE\_BD commands, but must not be written during BDM operation. Most users will only be interested in the STATUS register at \$FF01; other registers are only for use by BDM firmware and logic.

The instruction register is discussed for two conditions: when a **hardware** command is executed and when a **firmware** command is executed.

#### INSTRUCTION — BDM Instruction Register (hardware command)

(BDM) \$FF00

	Bit 7	6	5	4	3	2	1	Bit 0
	H/F	DATA	R/W	BKGND	W/B	BD/U	0	0
RESET:	0	0	0	0	0	0	0	0

The bits in the BDM instruction register have the following meanings when a **hardware** command is executed.

H/F — Hardware/Firmware Flag  
 0 = Firmware instruction  
 1 = Hardware instruction

DATA — Data Flag  
 0 = No data  
 1 = Data included in command

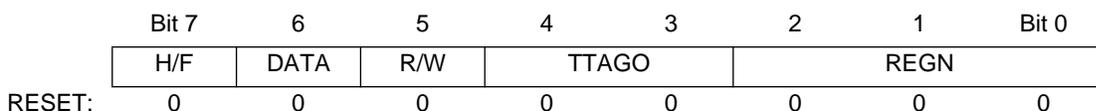
R/W — Read/Write Flag  
 0 = Write  
 1 = Read

BKGND — Hardware request to enter active background mode  
 0 = Not a hardware background command  
 1 = Hardware background command (INSTRUCTION = \$90)

W/B — Word/Byte Transfer Flag  
 0 = Byte transfer  
 1 = Word transfer

BD/U — BDM Map/User Map Flag  
 Indicates whether BDM registers and ROM are mapped to addresses \$FF00 to \$FFFF in the standard 64-Kbyte address space. Used only by hardware read/write commands.  
 0 = BDM resources not in map  
 1 = BDM resources in map

**INSTRUCTION** — BDM Instruction Register (firmware command) **(BDM) \$FF00**



The bits in the BDM instruction register have the following meanings when a **firmware** command is executed.

H/F — Hardware/Firmware Flag  
 0 = Firmware control logic  
 1 = Hardware control logic

DATA — Data Flag  
 0 = No data  
 1 = Data included in command

R/W — Read/Write Flag  
 0 = Write  
 1 = Read

TTAGO — Trace, Tag, Go Field

**Table 38 TTAGO Decoding**

TTAGO Value	Instruction
00	—
01	GO
10	TRACE1
11	TAGGO

REGN — Register/Next Field  
 Indicates which register is being affected by a command. In the case of a READ\_NEXT or WRITE\_NEXT command, index register X is pre-incremented by 2 and the word pointed to by X is then read or written.

**Table 39 REGN Decoding**

REGN Value	Instruction
000	—
001	—
010	READ/WRITE NEXT
011	PC
100	D
101	X
110	Y
111	SP

**STATUS** — BDM Status Register

**(BDM) \$FF01**

	Bit 7	6	5	4	3	2	1	Bit 0
	ENBDM	BDMACT	ENTAG	SDV	TRACE	0	0	0
RESET:	0	0	0	0	0	0	0	0
Sp Sing Chip & Peripheral	1	0	0	0	0	0	0	0

This register can be read or written by BDM commands or firmware.

**ENBDM** — Enable BDM (permit active background debug mode)  
 0 = BDM cannot be made active (hardware commands still allowed)  
 1 = BDM can be made active to allow firmware commands

**BDMACT** — Background Mode Active Status  
 0 = BDM not active  
 1 = BDM active and waiting for serial commands

**ENTAG** — Instruction Tagging Enable  
 Set by the TAGGO instruction and cleared when BDM is entered.  
 0 = Tagging not enabled, or BDM active  
 1 = Tagging active (BDM cannot process serial commands while tagging is active.)

**SDV** — Shifter Data Valid  
 Shows that valid data is in the serial interface shift register. Used by firmware-based instructions.  
 0 = No valid data  
 1 = Valid Data

**TRACE** — Asserted by the TRACE1 instruction

**SHIFTER** — BDM Shift Register

**(BDM) \$FF02, \$FF03**

Bit 15	14	13	12	11	10	9	Bit 8
S15	S14	S13	S12	S11	S10	S9	S8
Bit 7	6	5	4	3	2	1	Bit 0
S7	S6	S5	S4	S3	S2	S1	S0

This 16-bit register contains data being received or transmitted via the serial interface.

**ADDRESS** — BDM Address Register**(BDM) \$FF04, \$FF05**

Bit 15	14	13	12	11	10	9	Bit 8
A15	A14	A13	A12	A11	A10	A9	A8
Bit 7	6	5	4	3	2	1	Bit 0
A7	A6	A5	A4	A3	A2	A1	A0

This 16-bit register is temporary storage for BDM hardware and firmware commands.

**CCRS AV** — BDM CCR Holding Register**(BDM) \$FF06**

Bit 7	6	5	4	3	2	1	Bit 0
CCR7	CCR6	CCR5	CCR4	CCR3	CCR2	CCR1	CCR0

This register preserves the content of the CPU12 CCR while BDM is active.

**16.3 Instruction Tagging**

The instruction queue and cycle-by-cycle CPU activity can be reconstructed in real time or from trace history that was captured by a logic analyzer. However, the reconstructed queue cannot be used to stop the CPU at a specific instruction, because execution has already begun by the time an operation is visible outside the MCU. A separate instruction tagging mechanism is provided for this purpose.

Executing the BDM TAGGO command configures two MCU pins for tagging. Tagging information is latched on the falling edge of ECLK along with program information as it is fetched. Tagging is allowed in all modes. Tagging is disabled when BDM becomes active and BDM serial commands cannot be processed while tagging is active.

$\overline{\text{TAGHI}}$  is a shared function of the BKGD pin.

$\overline{\text{TAGLO}}$  is a shared function of the PE3/ $\overline{\text{LSTRB}}$  pin, a multiplexed I/O pin. For 1/4 cycle before and after the rising edge of the E clock, this pin is the  $\overline{\text{LSTRB}}$  driven output.

$\overline{\text{TAGLO}}$  and  $\overline{\text{TAGHI}}$  inputs are captured at the falling edge of the E clock. A logic zero on  $\overline{\text{TAGHI}}$  and/or  $\overline{\text{TAGLO}}$  marks (tags) the instruction on the high and/or low byte of the program word that was on the data bus at the same falling edge of the E clock.

The tag follows the information in the queue as the queue is advanced. When a tagged instruction reaches the head of the queue, the CPU enters active background debugging mode rather than executing the instruction. This is the mechanism by which a development system initiates hardware breakpoints.

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 which may be provided in Motorola data sheets and/or specifications 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.

MCUinit, MCUasm, MCUdebug, and RTEK are trademarks of Motorola, Inc. MOTOROLA and ! are registered trademarks of Motorola, Inc. Motorola, Inc. is an Equal Opportunity/Affirmative Action Employer.

**How to reach us:**

**USA/EUROPE/Locations Not Listed:** Motorola Literature Distribution;

P.O. Box 5405, Denver Colorado 80217. 1-800-441-2447, (303) 675-2140

**Mfax™:** RMFAX0@email.sps.mot.com - TOUCHTONE (602) 244-6609, U.S. and Canada Only 1-800-774-1848

**INTERNET:** <http://motorola.com/sps>

**JAPAN:** Nippon Motorola Ltd., Strategic Planning Office,

4-32-1, Nishi-Gotanda, Shinagawa-ku, Tokyo 141, Japan. 81-3-5487-8488

**ASIA PACIFIC:** Motorola Semiconductors H.K. Ltd.; 8B Tai Ping Industrial Park,  
51 Ting Kok Road, Tai Po, N.T., Hong Kong. 852-26629298

Mfax is a trademark of Motorola, Inc.



**MOTOROLA**