HC11

MC68HC11KW1

TECHNICAL
DATA

freescale™
semiconductor

MC68HC11KW1

High-density complementary metal oxide semiconductor (HCMOS) microcontroller unit

Conventions

Where abbreviations are used in the text, an explanation can be found in the glossary, at the back of this manual. Register and bit mnemonics are defined in the paragraphs describing them.

An overbar is used to designate an active-low signal, eg: $\overline{\text{RESET}}$.

Because the bits in any one register are not necessarily linked by a common function, the description of a register may appear in several sections referring to different aspects of device operation. A full description of a bit is given only in a section in which it has relevance. Elsewhere, it appears shaded in the register diagram and is only briefly described.

When the state of a bit on reset is described as ‘x’, this means that its state depends on factors such as the operating mode selected. A ‘u’ indicates that the bit’s state on reset is undefined.

TABLE OF CONTENTS

Paragraph Number | TITLE | Page Number

3
CENTRAL PROCESSING UNIT

4
OPERATING MODES AND ON-CHIP MEMORY

5
RESETS AND INTERRUPTS

6
PARALLEL INPUT/OUTPUT

7
SERIAL COMMUNICATIONS INTERFACE

8
SERIAL PERIPHERAL INTERFACE

9
TIMING SYSTEM

10
ANALOG-TO-DIGITAL CONVERTER

A
ELECTRICAL SPECIFICATIONS

B
MECHANICAL DATA

C
DEVELOPMENT SYSTEMS

LIST OF FIGURES

LIST OF TABLES

1
INTRODUCTION

The MC68HC11KW1 8-bit microcontroller is a member of the M68HC11 family of HCMOS microcontrollers. It has 640 bytes of EEPROM and 768 bytes of RAM. Making use of a 100-pin TQFP package, a non-multiplexed expanded bus is a feature of this device. The main timer system includes three input captures, four output compares and a software selectable input capture or output compare. There are two additional 16-bit timers, each with three output compares and one software selectable input capture or output compare. Other major features of this device are: a 10-channel, 10-bit resolution A/D converter, four PWM timer channels, an SPI (serial peripheral interface) and an enhanced SCI (serial communications interface). In common with other family members, the MC68HC11KW1 also includes an 8-bit pulse accumulator circuit, a real time interrupt facility, and a computer operating properly watchdog system. This device is intended for use in expanded memory applications.

1.1 Features

- Low power, high performance M68HC11 CPU core with 4MHz internal bus frequency
- 768 bytes of RAM
- 640 bytes of byte-erasable EEPROM, with on-chip charge pump
- 448 bytes of boot ROM
- Up to 70 general purpose I/O lines, plus up to 10 input-only lines
- Non-multiplexed address and data buses, permitting direct access to the full 64K byte address map
- Memory expansion unit, with six address extension lines, allowing up to (for example) sixteen 32K byte banks of external memory to be addressed in either of two bank windows
- Four external chip selects
- 16-bit timer with 3/4 input captures and 5/4 output compares; pulse accumulator and COP watchdog timer
- Real-time interrupt circuit

- Two additional 16-bit timers, each with 3 output compares and one input capture or output compare (may be externally clocked, if required, for external event counter operation)
- SCI subsystem (NRZ type for compatibility with standard RS232 systems) with parity and a modulus prescaler
- SPI subsystem, with software selectable MSB/LSB first option and increased baud rate selection range
- 10-channel, 10-bit analog-to-digital converter
- Four 8-bit PWM timer channels
- Available in 100-pin TQFP package

1.2 Mask option

There is a single mask option on the MC68HC11KW1, which is programmed during manufacture and must be specified on the order form:

- Security option (available/unavailable). See Section 4.6.3

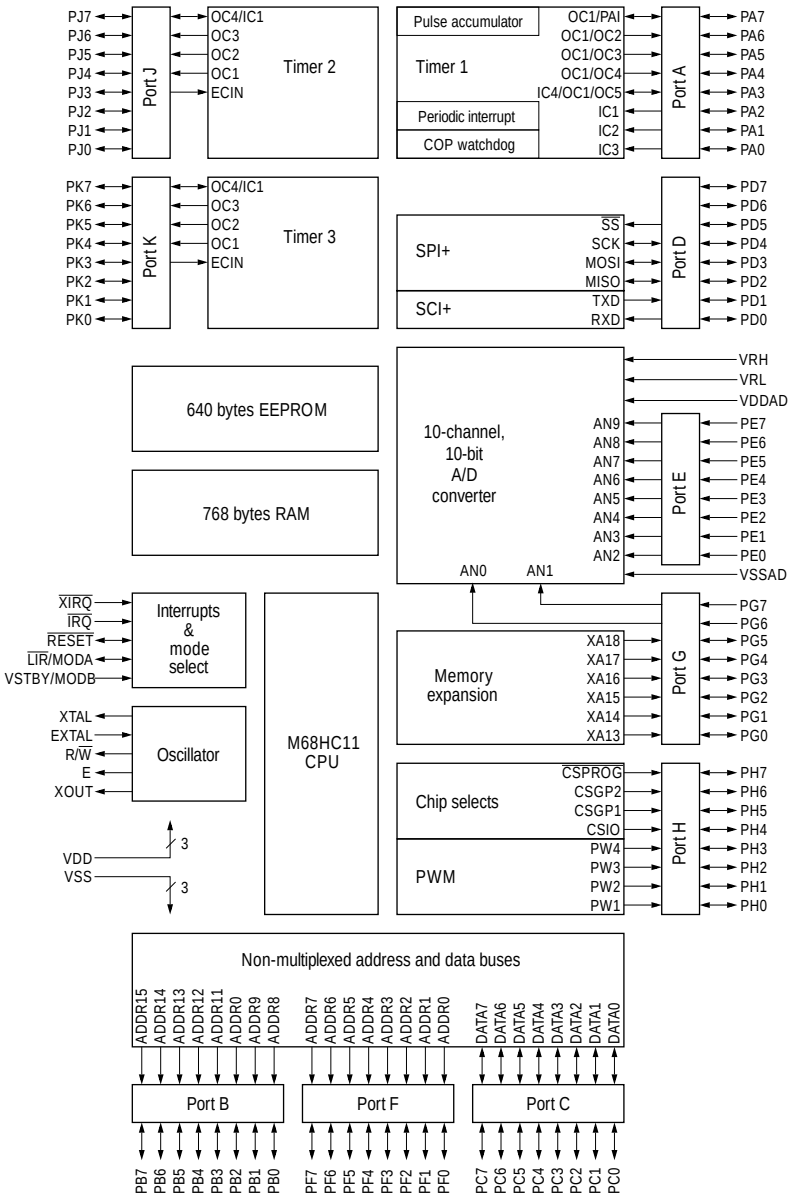


Figure 1-1 MC68HC11KW1 block diagram

THIS PAGE LEFT BLANK INTENTIONALLY

2
PIN DESCRIPTIONS

The MC68HC11KW1 is available packaged in a 100-pin thin quad flat pack (TQFP), as shown in Figure 2-1. Most pins on this MCU serve two or more functions, as described in the following paragraphs.

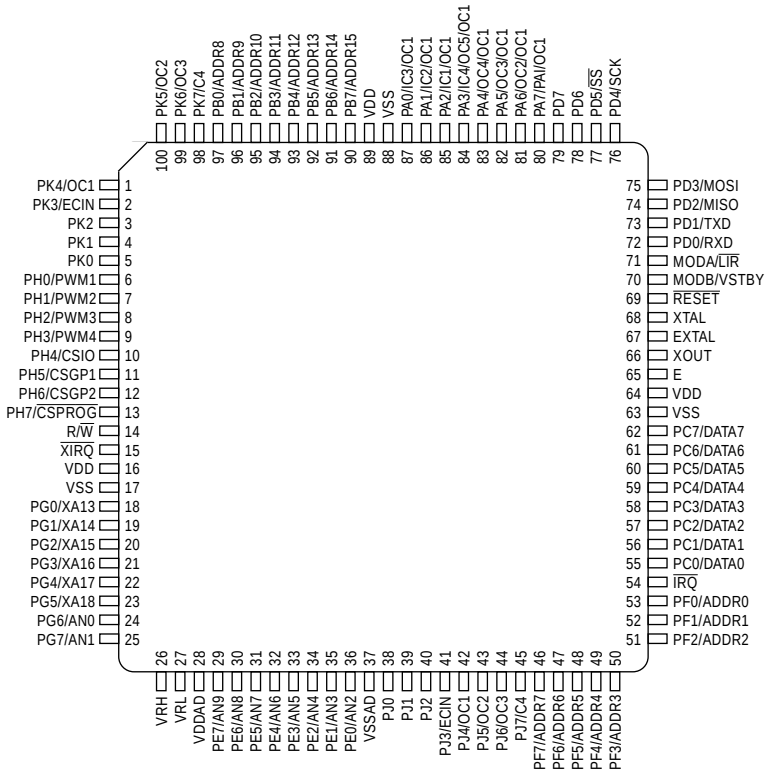


Figure 2-1 MC68HC11KW1 100-pin TQFP

2.1 $\overline{\text{RESET}}$

An active-low, bidirectional control signal, $\overline{\text{RESET}}$ acts as an input to initialize the MCU to a known start-up state. It also acts as an open-drain output to indicate that an internal failure has been detected in either the clock monitor or the COP watchdog circuit. The CPU distinguishes between internal and external reset conditions by sensing whether the reset pin rises to a logic one in less than four E clock cycles after an internal reset has been released. It is therefore not advisable to connect an external resistor-capacitor (RC) power-up delay circuit to the reset pin of M68HC11 devices because the circuit charge time constant can cause the device to misinterpret the type of reset that occurred. Refer to Section 5 for further information.

Figure 2-2 illustrates a typical reset circuit that includes an external switch together with a low voltage inhibit circuit, to prevent power transitions, or RAM or EEPROM corruption.

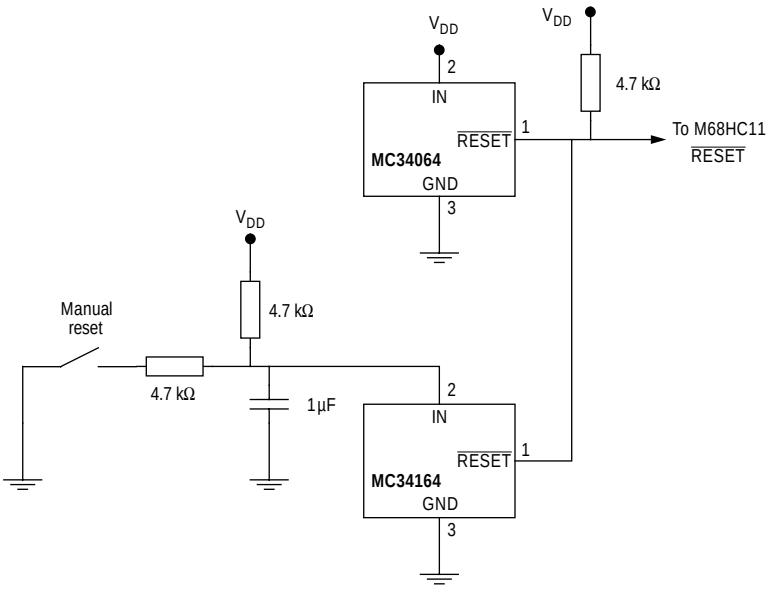


Figure 2-2 External reset circuitry

2.2 Crystal driver and external clock input (XTAL, EXTAL)

These two pins provide the interface for either a crystal or a CMOS compatible clock to control the internal clock generator circuitry. The frequency applied to these pins must be four times higher than the desired E clock rate. Refer to Figure 2-3.

The XTAL pin is normally left unconnected when an external CMOS compatible clock input is connected to the EXTAL pin. The XTAL output is normally intended to drive only a crystal. The XTAL output can be buffered with a high-impedance buffer, or it can be used to drive the EXTAL input of another M68HC11 family device.

In all cases, use caution when designing circuitry associated with the oscillator pins.

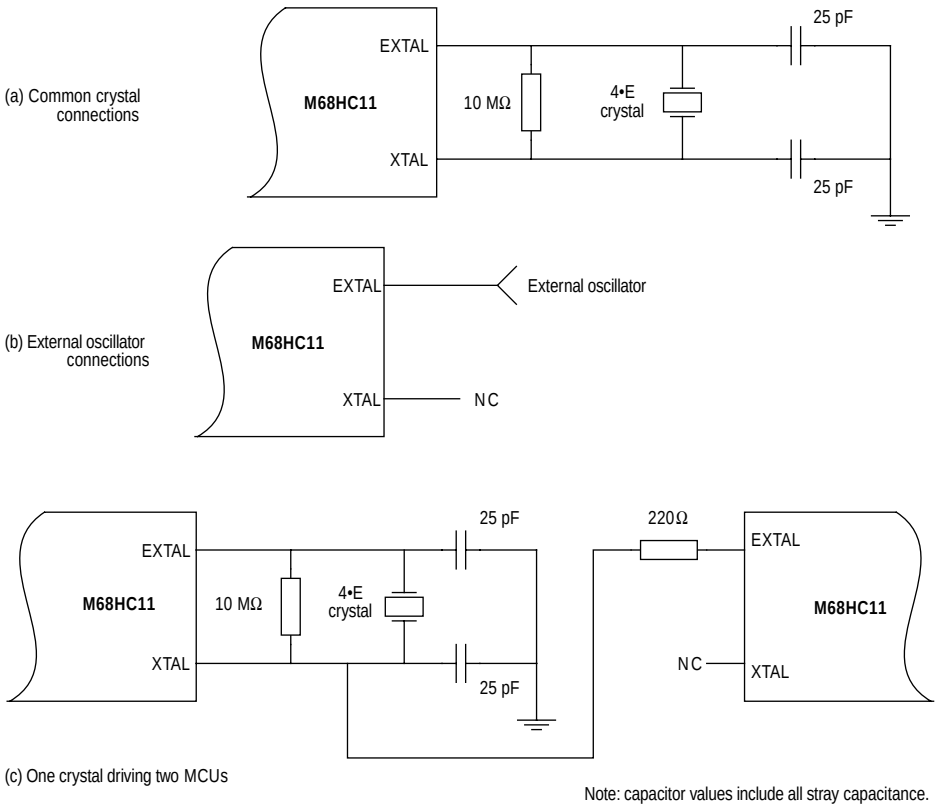


Figure 2-3 Oscillator connections

2.3 VDD and VSS

Power is supplied to the microcontroller via these pins. VDD is the positive supply and VSS is ground. The MCU operates from a 5V (nominal) power supply.

It is in the nature of CMOS designs that very fast signal transitions occur on the MCU pins. These short rise and fall times place very high short-duration current demands on the power supply. To prevent noise problems, special care must be taken to provide good power supply bypassing at the MCU. Bypass capacitors should have good high-frequency characteristics and be as close to the MCU as possible. Bypassing requirements vary, depending on how heavily the MCU pins are loaded.

The MC68HC11KW1 has four VDD pins and four VSS pins. One pair of these pins is reserved for supplying power to the analog-to-digital converter (VDDAD, VSSAD); the remaining pins are used for the internal logic, and to supply power to the port logic on either half of the chip.

2.4 E clock output (E)

E is the output connection for the internally generated E clock. The signal from E is used as a timing reference. The frequency of the E clock output is one quarter that of the input frequency at the XTAL and EXTAL pins. When E clock output is low, an internal process is taking place; when it is high, data is being accessed. All clocks, including the E clock, are halted when the MCU is in STOP mode. The E clock output can be turned off in single-chip modes to reduce the effects of RFI (see Section 4.3.2.5).

2.5 XOUT

The XOUT pin outputs the buffered CLKX signal, if enabled by the XCLK bit in the CONFIG register. The frequency of CLKX can be selected using two bits in the OPT2 register (XDV1 and XDV2). On reset, CLKX has the same frequency as EXTAL (4E). See Section 4.

Note that the phase relationship between CLKX and EXTAL cannot be predicted.

2.6 Interrupt request ($\overline{\text{IRQ}}$)

The $\overline{\text{IRQ}}$ input provides a means of applying asynchronous interrupt requests to the MCU. Either falling-edge-sensitive triggering or level-sensitive triggering is program selectable (OPTION register). $\overline{\text{IRQ}}$ is always configured to level-sensitive-triggering at reset.

Note: Connect an external pull-up resistor, typically 4.7 kΩ, to VDD when $\overline{\text{IRQ}}$ is used in a level sensitive wired-OR configuration. See also Section 2.7.

2.7 Nonmaskable interrupt ($\overline{\text{XIRQ}}$)

The $\overline{\text{XIRQ}}$ input provides a means of requesting a nonmaskable interrupt after reset initialization. During reset, the X bit in the condition code register (CCR) is set and any interrupt is masked until MCU software enables it. $\overline{\text{XIRQ}}$ is often used as a power loss detect interrupt.

Whenever $\overline{\text{XIRQ}}$ or $\overline{\text{IRQ}}$ is used with multiple interrupt sources ($\overline{\text{IRQ}}$ must be configured for level-sensitive operation if there is more than one source of interrupt), each source must drive the interrupt input with an open-drain type of driver to avoid contention between outputs. There should be a single pull-up resistor near the MCU interrupt input pin (typically 4.7 kΩ). There must also be an interlock mechanism at each interrupt source so that the source holds the interrupt line low until the MCU recognizes and acknowledges the interrupt request. If one or more interrupt source is still pending after the MCU services a request, the interrupt line will still be held low and the MCU will be interrupted again as soon as the interrupt mask bit in the MCU is cleared (normally upon return from an interrupt). Refer to Section 5.

2.8 MODA and MODB (MODA/$\overline{\text{LIR}}$ and MODB/VSTBY)

During reset, MODA and MODB select one of the four operating modes. Refer to Section 4.

After the operating mode has been selected, the $\overline{\text{LIR}}$ pin provides an open-drain output (driven low) to indicate that execution of an instruction has begun. In order to detect consecutive instructions in a high-speed application, this signal drives high for a short time to prevent false triggering. A series of E clock cycles occurs during execution of each instruction. The $\overline{\text{LIR}}$ signal goes low during the first E clock cycle of each instruction (opcode fetch). This output is provided for assistance in program debugging, and its operation is controlled by the LIRDV bit in the OPT2 register.

The VSTBY pin is used to input RAM stand-by power. The MCU is powered from the VDD pin unless the difference between the level of V_{STBY} and V_{DD} is greater than one MOS threshold (about 0.7 volts). When these voltages differ by more than 0.7 volts, the internal RAM and part of the reset logic are powered from V_{STBY} rather than V_{DD}. This allows RAM contents to be retained without VDD power applied to the MCU. Reset must be driven low before V_{DD} is removed and must remain low until V_{DD} has been restored to a valid level.

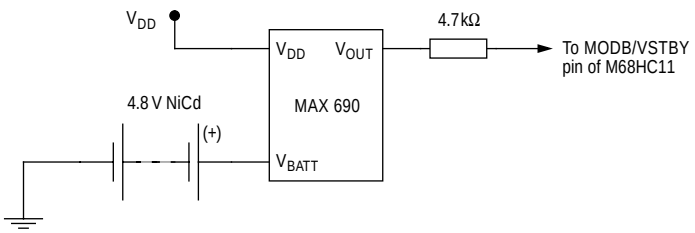


Figure 2-4 RAM stand-by connections

2.9 VRH and VRL

These pins provide the reference voltages for the analog-to-digital converter.

2.10 R/$\overline{W}$

In expanded and test modes, R/$\overline{W}$ performs the read/write function. R/$\overline{W}$ signals the direction of transfers on the external data bus. A high on this pin indicates that a read cycle is in progress.

In single chip mode the R/$\overline{W}$ signal is driven low.

2.11 Port signals

The MC68HC11KW1 includes 80 pins that are arranged into ten 8-bit ports (A, B, C, D, E, F, G, H, J and K). All the port pins are bidirectional, except for PG7, PG6 and port E pins [7:0]; these are input only. Most of the bidirectional ports serve a purpose other than I/O, depending on the operating mode or peripheral function selected. The input-only pins may be used as general-purpose inputs, or as inputs to the A/D converter. Note that ports B, C, and F are available for I/O functions only in single chip and bootstrap modes. Refer to Table 2-1 for details of the port signals' functions in different operating modes.

Note: When using the information about port functions, do not confuse pin function with the electrical state of the pin at reset. All general purpose I/O pins configured as inputs at reset are in a high-impedance state. Port data registers reflect the functional state of the port at reset. The pin function is mode dependent.

2.11.1 Port A

Port A is an 8-bit, general-purpose I/O port with a data register (PORTA) and a data direction register (DDRA). Port A pins share functions with the main 16-bit timer system, Timer 1 (see Section 9 for further information). PORTA can be read at any time and always returns the pin level. If written, PORTA stores the data in internal latches. The pins are driven only if they are configured as outputs. Writes to PORTA do not change the pin state when the pins are configured for timer output compares.

Out of reset, port A pins [7:0] are general purpose high-impedance inputs. When the functions associated with these pins are disabled, the bits in DDRA govern the I/O state of the associated pin. For further information, refer to Section 6.

Table 2-1 Port signal functions

Port/bit	Single chip and bootstrap mode	Expanded multiplexed and special test mode
PA7	PA7/PAI and/or OC1	
PA6	PA6/OC2 and/or OC1	
PA5	PA5/OC3 and/or OC1	
PA4	PA4/OC4 and/or OC1	
PA3	PA3/OC5/IC4 and/or OC1	
PA2	PA2/IC1	
PA1	PA1/IC2	
PA0	PA0/IC3	
PB[7:0]	PB[7:0]	ADDR[15:8]
PC[7:0]	PC[7:0]	DATA[7:0]
PD[7,6]	PD[7, 6]	
PD5	PD5/$\overline{SS}$	
PD4	PD4/SCK	
PD3	PD3/MOSI	
PD2	PD2/MISO	
PD1	PD1/TXD	
PD0	PD0/RXD	
PE[7:0]	Input only / analog inputs	
PF[7:0]	PF[7:0]	ADDR[7:0]
PG[7,6]	Input only / analog inputs	
PG[5:0]	PG[5:0]	PG[5:0] / XA[18:13]
PH7	PH7	PH7 / $\overline{CSPROG}$
PH6	PH6	PH6 / CSGP2
PH5	PH5	PH5 / CSGP1
PH4	PH4	PH4 / CSIO
PH[3:0]	PH[3:0] / PWM[4:1]	
PJ7	PJ7 / C4	
PJ6	PJ6 / OC3	
PJ5	PJ5 / OC2	
PJ4	PJ4 / OC1	
PJ3	PJ3 / ECIN	
PJ[2,0]	PJ[2,0]	
PK7	PK7 / C4	
PK6	PK6 / OC3	
PK5	PK5 / OC2	
PK4	PK4 / OC1	
PK3	PK3 / ECIN	
PK[2,0]	PK[2,0]	

2.11.2 Port B

Port B is an 8-bit, general-purpose I/O port with a data register (PORTB) and a data direction register (DDRB). In single chip mode, port B pins are general purpose I/O pins (PB[7:0]). In expanded mode, port B pins act as the high-order address lines (ADDR[15:8]) of the address bus.

PORTB can be read at any time and always returns the pin level. If PORTB is written, the data is stored in internal latches. The pins are driven only if they are configured as outputs in single chip or bootstrap mode. For further information, refer to Section 6.

Port B pins include on-chip pull-up devices which can be enabled or disabled via the port pull-up assignment register (PPAR).

2.11.3 Port C

Port C is an 8-bit, general-purpose I/O port with a data register (PORTC) and a data direction register (DDRC). In single chip mode, port C pins are general purpose I/O pins (PC[7:0]). In the expanded mode, port C pins are configured as data bus pins (DATA[7:0]).

PORTC can be read at any time; inputs return the pin level and outputs return the pin driver input level. If PORTC is written, the data is stored in internal latches. The pins are driven only if they are configured as outputs in single chip or bootstrap mode. Port C pins are general purpose inputs out of reset in single chip and bootstrap modes. In expanded and test modes, these pins are data bus lines out of reset.

The CWOM control bit in the OPT2 register disables port C's p-channel output drivers. Because the n-channel driver is not affected by CWOM, setting CWOM causes port C to become an open-drain-type output port suitable for wired-OR operation. In wired-OR mode (PORTC bits at logic level zero), the pins are actively driven low by the n-channel driver. When a port C bit is at logic level one, the associated pin is in a high impedance state as neither the n-channel nor the p-channel devices are active. It is customary to have an external pull-up resistor on lines that are driven by open-drain devices. Port C can only be configured for wired-OR operation when the MCU is in single chip mode. For further information, refer to Section 6.

2.11.4 Port D

Port D, an 8-bit, general-purpose I/O port, has a data register (PORTD) and a data direction register (DDRD). All the port D pins can be used for general purpose I/O, and pins [5:0] can also be used for the serial peripheral interface (SPI, pins [5:2]) and the serial communications interface (SCI, pins [1,0]).

PORTD can be read at any time; inputs return the pin level and outputs return the pin driver input level. If PORTD is written, the data is stored in internal latches. The pins are driven only if port D is configured for general purpose output.

The DWOM bit in SPCR disables the p-channel output drivers of pins D[5:2], and the WOMS bit in SCCR1 disables those of pins D[1,0]. Because the n-channel driver is not affected by DWOM or WOMS, setting either bit causes the corresponding port D pins to become open-drain-type outputs suitable for wired-OR operation. In wired-OR mode (PORTD bits at logic level zero), the pins are actively driven low by the n-channel driver. When a port D bit is at logic level one, the associated pin is in a high impedance state as neither the n-channel nor the p-channel devices are active. It is customary to have an external pull-up resistor on lines that are driven by open-drain devices. Port D can be configured for wired-OR operation when the MCU is in single chip or expanded mode.

For further information, refer to Section 6, Section 7 (SCI) and Section 8 (SPI).

2.11.5 Port E

Port E pins can be used as the analog inputs for the analog-to-digital converter, or as general-purpose inputs. For further information, refer to Section 6 and Section 10 (A/D).

2.11.6 Port F

Port F is an 8-bit, general-purpose I/O port with a data register (PORTF) and a data direction register (DDRF). In single chip mode, port F pins are general purpose I/O pins (PF[7:0]). In expanded mode, port F pins act as the low-order address lines (ADDR[7:0]) of the address bus.

PORTF can be read at any time and always returns the pin level. If PORTF is written, the data is stored in internal latches. The pins are driven only if they are configured as outputs in single chip or bootstrap mode.

Port F pins include on-chip pull-up devices that can be enabled or disabled via the port pull-up assignment register (PPAR). For further information, refer to Section 6.

2.11.7 Port G

In normal modes, Port G is an 8-bit general-purpose port with 6 I/O lines (PG[5:0]), and two input only lines (PG[7, 6]). Associated with port G are a data register (PORTG), a data direction register (DDRG) and an assignment register (PGAR). Pins [7, 6] can be used as general-purpose inputs, or as inputs to the analog-to-digital converter. The functions of pins [5:0] are controlled by bits in the port G assignment register (PGAR), which select whether the pins are used for general purpose I/O, or, in expanded mode, for the memory expansion lines XA[18:13].

PORTG can be read at any time and always returns the pin level. If PORTG is written, the data is stored into an internal latch. The pin is driven only if it is configured as an output.

Pins [5:0] have on-chip pull-up devices that can be enabled or disabled via the port pull-up assignment register (PPAR). Refer to Section 6, Section 10 (A/D) and Section 4.

2.11.8 Port H

Port H is an 8-bit general-purpose I/O port with a data register (PORTH) and a data direction register (DDRH). Port H lines can be used for general-purpose I/O, for chip select lines (PH[7:4]), and for the pulse width modulation timer (PWM, PH[3:0]).

PORTH can be read at any time and always returns the pin level. If PORTH is written, the data is stored into an internal latch. The pin is driven only if it is configured as an output.

Port H pins include on-chip pull-up devices that can be enabled or disabled via the port pull-up assignment register (PPAR). For further information, refer to Section 6, Section 9 (Timing system) and Section 4.

2.11.9 Port J

Port J is an 8-bit, general-purpose I/O port with a data register (PORTJ) and a data direction register (DDRJ). Port J lines can be used for general-purpose I/O, and pins [7:3] share functions with one of the 16-bit timers, Timer 2.

PORTJ can be read at any time and always returns the pin level. If written, PORTJ stores the data in internal latches. The pins are driven only if they are configured as outputs. Writes to PORTJ do not change the pin state when the pins are configured for timer output compares.

Out of reset, port J pins [7:0] are general purpose high-impedance inputs. When the functions associated with these pins are disabled, the bits in DDRJ govern the I/O state of the associated pin. For further information, refer to Section 6 and Section 9 (Timing system).

2.11.10 Port K

Port K is an 8-bit general-purpose I/O port with a data register (PORTK) and a data direction register (DDRK). Port K lines can be used for general-purpose I/O, and pins [7:3] share functions with one of the 16-bit timers, Timer 3.

PORTK can be read at any time and always returns the pin level. If written, PORTK stores the data in internal latches. The pins are driven only if they are configured as outputs. Writes to PORTK do not change the pin state when the pins are configured for timer output compares.

Out of reset, port K pins [7:0] are general purpose high-impedance inputs. When the functions associated with these pins are disabled, the bits in DDRK govern the I/O state of the associated pin. For further information, refer to Section 6 and Section 9 (Timing system).

3
CENTRAL PROCESSING UNIT

This section discusses the M68HC11 central processing unit (CPU) architecture, its addressing modes and the instruction set. For more detailed information on the instruction set, refer to the ***M68HC11 Reference Manual (M68HC11RM/AD)***.

The CPU is designed to treat all peripheral, I/O and memory locations identically, as addresses in the 64Kbyte memory map. This is referred to as memory-mapped I/O. There are no special instructions for I/O that are separate from those used for memory. This architecture also allows accessing an operand from an external memory location with no execution-time penalty.

3.1 Registers

M68HC11 CPU registers are an integral part of the CPU and are not addressed as if they were memory locations. The seven registers are shown in Figure 3-1 and are discussed in the following paragraphs.

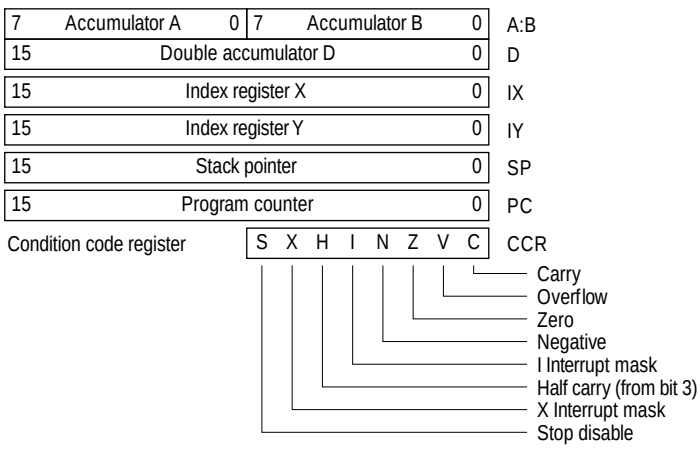


Figure 3-1 Programming model

3.1.1 Accumulators A, B and D

Accumulators A and B are general purpose 8-bit registers that hold operands and results of arithmetic calculations or data manipulations. For some instructions, these two accumulators are treated as a single double-byte (16-bit) accumulator called accumulator D. Although most operations can use accumulators A or B interchangeably, the following exceptions apply:

- The ABX and ABY instructions add the contents of 8-bit accumulator B to the contents of 16-bit register X or Y, but there are no equivalent instructions that use A instead of B.
- The TAP and TPA instructions transfer data from accumulator A to the condition code register, or from the condition code register to accumulator A, however, there are no equivalent instructions that use B rather than A.
- The decimal adjust accumulator A (DAA) instruction is used after binary-coded decimal (BCD) arithmetic operations, but there is no equivalent BCD instruction to adjust accumulator B.
- The add, subtract, and compare instructions associated with both A and B (ABA, SBA, and CBA) only operate in one direction, making it important to plan ahead to ensure the correct operand is in the correct accumulator.

3.1.2 Index register X (IX)

The IX register provides a 16-bit indexing value that can be added to the 8-bit offset provided in an instruction to create an effective address. The IX register can also be used as a counter or as a temporary storage register.

3.1.3 Index register Y (IY)

The 16-bit IY register performs an indexed mode function similar to that of the IX register. However, most instructions using the IY register require an extra byte of machine code and an extra cycle of execution time because of the way the opcode map is implemented. Refer to Section 3.3 for further information.

3.1.4 Stack pointer (SP)

The M68HC11 CPU has an automatic program stack. This stack can be located anywhere in the address space and can be any size up to the amount of memory available in the system. Normally the SP is initialized by one of the first instructions in an application program. The stack is configured as a data structure that grows downward from high memory to low memory. Each time a new byte is pushed onto the stack, the SP is decremented. Each time a byte is pulled from the stack, the SP is incremented. At any given time, the SP holds the 16-bit address of the next free location in the stack. Figure 3-2 is a summary of SP operations.

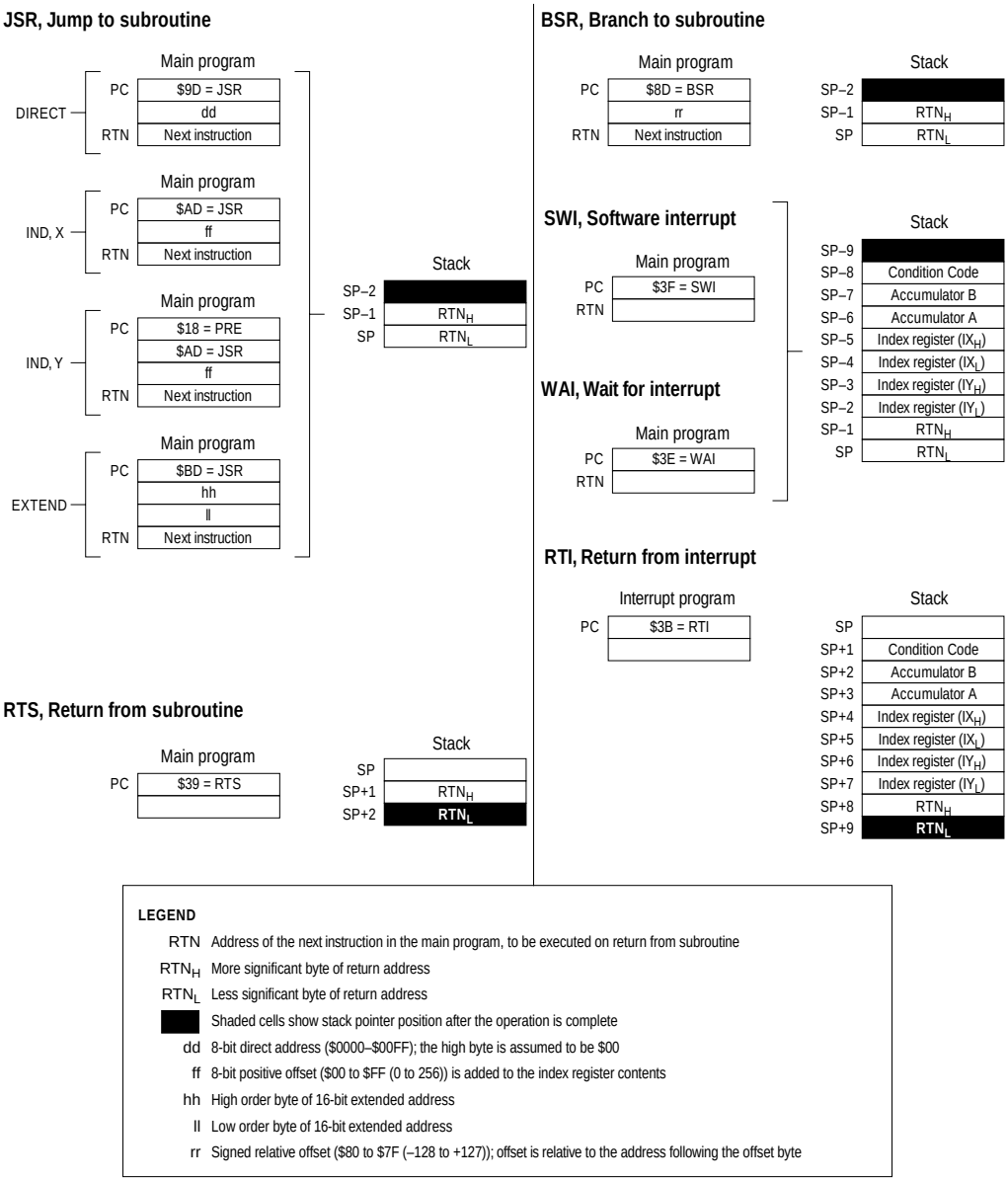


Figure 3-2 Stacking operations

When a subroutine is called by a jump to subroutine (JSR) or branch to subroutine (BSR) instruction, the address of the instruction after the JSR or BSR is automatically pushed onto the stack, less significant byte first. When the subroutine is finished, a return from subroutine (RTS) instruction is executed. The RTS pulls the previously stacked return address from the stack, and loads it into the program counter. Execution then continues at this recovered return address.

When an interrupt is recognized, the current instruction finishes normally, the return address (the current value in the program counter) is pushed onto the stack, all of the CPU registers are pushed onto the stack, and execution continues at the address specified by the vector for the interrupt. At the end of the interrupt service routine, an RTI instruction is executed. The RTI instruction causes the saved registers to be pulled off the stack in reverse order. Program execution resumes at the return address.

There are instructions that push and pull the A and B accumulators and the X and Y index registers. These instructions are often used to preserve program context. For example, pushing accumulator A onto the stack when entering a subroutine that uses accumulator A, and then pulling accumulator A off the stack just before leaving the subroutine, ensures that the contents of a register will be the same after returning from the subroutine as it was before starting the subroutine.

3.1.5 Program counter (PC)

The program counter, a 16-bit register, contains the address of the next instruction to be executed. After reset, the program counter is initialized from one of six possible vectors, depending on operating mode and the cause of reset.

Table 3-1 Reset vector comparison

	POR or RESET pin	Clock monitor	COP watchdog
Normal	$FFFE, $FFFF	$FFFC, $FFFD	$FFFA, $FFFB
Test or Boot	$BFFE, $BFFF	$BFFE, $BFFF	$BFFE, $BFFF

3.1.6 Condition code register (CCR)

This 8-bit register contains five condition code indicators (C, V, Z, N, and H), two interrupt masking bits, (IRQ and XIRQ) and a stop disable bit (S). In the M68HC11 CPU, condition codes are automatically updated by most instructions. For example, load accumulator A (LDAA) and store accumulator A (STAA) instructions automatically set or clear the N, Z, and V condition code flags. Pushes, pulls, add B to X (ABX), add B to Y (ABY), and transfer/exchange instructions do not affect the condition codes. Refer to Table 3-2, which shows the condition codes that are affected by a particular instruction.

3.1.6.1 Carry/borrow (C)

The C-bit is set if the arithmetic logic unit (ALU) performs a carry or borrow during an arithmetic operation. The C-bit also acts as an error flag for multiply and divide operations. Shift and rotate instructions operate with and through the carry bit to facilitate multiple-word shift operations.

3.1.6.2 Overflow (V)

The overflow bit is set if an operation causes an arithmetic overflow. Otherwise, the V-bit is cleared.

3.1.6.3 Zero (Z)

The Z-bit is set if the result of an arithmetic, logic, or data manipulation operation is zero. Otherwise, the Z-bit is cleared. Compare instructions do an internal implied subtraction and the condition codes, including Z, reflect the results of that subtraction. A few operations (INX, DEX, INY, and DEY) affect the Z-bit and no other condition flags. For these operations, only = and ≠ conditions can be determined.

3.1.6.4 Negative (N)

The N-bit is set if the result of an arithmetic, logic, or data manipulation operation is negative; otherwise, the N-bit is cleared. A result is said to be negative if its most significant bit (MSB) is set (MSB = 1). A quick way to test whether the contents of a memory location has the MSB set is to load it into an accumulator and then check the status of the N-bit.

3.1.6.5 Interrupt mask (I)

The interrupt request (IRQ) mask (I-bit) is a global mask that disables all maskable interrupt sources. While the I-bit is set, interrupts can become pending, but the operation of the CPU continues uninterrupted until the I-bit is cleared. After any reset, the I-bit is set by default and can only be cleared by a software instruction. When an interrupt is recognized, the I-bit is set after the registers are stacked, but before the interrupt vector is fetched. After the interrupt has been serviced, a return from interrupt instruction is normally executed, restoring the registers to the values that were present before the interrupt occurred. Normally, the I-bit is zero after a return from interrupt is executed. Although the I-bit can be cleared within an interrupt service routine, ‘nesting’ interrupts in this way should only be done when there is a clear understanding of latency and of the arbitration mechanism. Refer to Section 5.

3.1.6.6 Half carry (H)

The H-bit is set when a carry occurs between bits 3 and 4 of the arithmetic logic unit during an ADD, ABA, or ADC instruction. Otherwise, the H-bit is cleared. Half carry is used during BCD operations.

3.1.6.7 X interrupt mask (X)

The XIRQ mask (X) bit disables interrupts from the $\overline{\text{XIRQ}}$ pin. After any reset, X is set by default and must be cleared by a software instruction. When an $\overline{\text{XIRQ}}$ interrupt is recognized, the X- and I-bits are set after the registers are stacked, but before the interrupt vector is fetched. After the interrupt has been serviced, an RTI instruction is normally executed, causing the registers to be restored to the values that were present before the interrupt occurred. The X interrupt mask bit is set only by hardware $\overline{\text{RESET}}$ or $\overline{\text{XIRQ}}$ acknowledge). X is cleared only by program instruction (TAP, where the associated bit of A is 0; or RTI, where bit 6 of the value loaded into the CCR from the stack has been cleared). There is no hardware action for clearing X.

3.1.6.8 Stop disable (S)

Setting the STOP disable (S) bit prevents the STOP instruction from putting the M68HC11 into a low-power stop condition. If the STOP instruction is encountered by the CPU while the S-bit is set, it is treated as a no-operation (NOP) instruction, and processing continues to the next instruction. S is set by reset — STOP disabled by default.

3.2 Data types

The M68HC11 CPU 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. Because the M68HC11 is an 8-bit CPU, there are no special requirements for alignment of instructions or operands.

3.3 Opcodes and operands

The M68HC11 family of microcontrollers 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 able to be expressed in 8-bit binary numbers.

A four-page opcode map has been implemented to expand the number of instructions. An additional byte, called a prebyte, directs the processor from page 0 of the opcode map to one of the other three pages. As its name implies, the additional byte precedes the opcode.

A complete instruction consists of a prebyte, if any, an opcode, and zero, one, two, or three operands. The operands contain information the CPU needs for executing the instruction. Complete instructions can be from one to five bytes long.

3.4 Addressing modes

Six addressing modes; immediate, direct, extended, indexed, inherent, and relative, detailed in the following paragraphs, can be used to access memory. All modes except inherent mode use an effective address. The effective address is the memory address from which the argument is fetched or stored, or the address from which execution is to proceed. The effective address can be specified within an instruction, or it can be calculated.

3.5 Immediate (IMM)

In the immediate addressing mode an argument is contained in the byte(s) immediately following the opcode. The number of bytes following the opcode matches the size of the register or memory location being operated on. There are two, three, and four (if prebyte is required) byte immediate instructions. The effective address is the address of the byte following the instruction.

3.5.1 Direct (DIR)

In the direct addressing mode, the low-order byte of the operand address is contained in a single byte following the opcode, and the high-order byte of the address is assumed to be $00. Addresses $00–$FF are thus accessed directly, using two-byte instructions. Execution time is reduced by eliminating the additional memory access required for the high-order address byte. In most applications, this 256-byte area is reserved for frequently referenced data. In M68HC11 MCUs, the memory map can be configured for combinations of internal registers, RAM, or external memory to occupy these addresses.

3.5.2 Extended (EXT)

In the extended addressing mode, the effective address of the argument is contained in two bytes following the opcode byte. These are three-byte instructions (or four-byte instructions if a prebyte is required). One or two bytes are needed for the opcode and two for the effective address.

3.5.3 Indexed (IND, X; IND, Y)

In the indexed addressing mode, an 8-bit unsigned offset contained in the instruction is added to the value contained in an index register (IX or IY) — the sum is the effective address. This addressing mode allows referencing any memory location in the 64Kbyte address space. These are two- to five-byte instructions, depending on whether or not a prebyte is required.

3.5.4 Inherent (INH)

In the inherent addressing mode, all the information necessary to execute the instruction is contained in the opcode. Operations that use only the index registers or accumulators, as well as control instructions with no arguments, are included in this addressing mode. These are one or two-byte instructions.

3.5.5 Relative (REL)

The relative addressing mode is used only for branch instructions. If the branch condition is true, an 8-bit signed offset included in the instruction is added to the contents of the program counter to form the effective branch address. Otherwise, control proceeds to the next instruction. These are usually two-byte instructions.

3.6 Instruction set

Refer to Table 3-2, which shows all the M68HC11 instructions in all possible addressing modes. For each instruction, the table shows the operand construction, the number of machine code bytes, and execution time in CPU E clock cycles.

Table 3-2 Instruction set (Page 1 of 6)

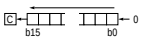
Mnemonic	Operation	Description	Addressing mode	Instruction			Condition codes							
				Opcode	Operand	Cycles	S	X	H	I	N	Z	V	C
ABA	Add accumulators	A + B ⇒ A	INH	1B	—	2	—	—	Δ	—	Δ	Δ	Δ	Δ
ABX	Add B to X	IX + (00:B) ⇒ IX	INH	3A	—	3	—	—	—	—	—	—	—	—
ABY	Add B to Y	IY + (00:B) ⇒ IY	INH	18 3A	—	4	—	—	—	—	—	—	—	—
ADCA (opr)	Add with carry to A	A + M + C ⇒ A	A IMM	89	ii	2	—	—	Δ	—	Δ	Δ	Δ	Δ
			A DIR	99	dd	3								
			A EXT	B9	hh ll	4								
			A IND, X	A9	ff	4								
			A IND, Y	18 A9	ff	5								
ADCB (opr)	Add with carry to B	B + M + C ⇒ B	B IMM	C9	ii	2	—	—	Δ	—	Δ	Δ	Δ	Δ
			B DIR	D9	dd	3								
			B EXT	F9	hh ll	4								
			B IND, X	E9	ff	4								
			B IND, Y	18 E9	ff	5								
ADDA (opr)	Add memory to A	A + M ⇒ A	A IMM	8B	ii	2	—	—	Δ	—	Δ	Δ	Δ	Δ
			A DIR	9B	dd	3								
			A EXT	BB	hh ll	4								
			A IND, X	AB	ff	4								
			A IND, Y	18 AB	ff	5								
ADDB (opr)	Add memory to B	B + M ⇒ B	B IMM	CB	ii	2	—	—	Δ	—	Δ	Δ	Δ	Δ
			B DIR	DB	dd	3								
			B EXT	FB	hh ll	4								
			B IND, X	EB	ff	4								
			B IND, Y	18 EB	ff	5								
ADDD (opr)	Add 16-bit to D	D + (M:M+1) ⇒ D	IMM	C3	jj kk	4	—	—	—	—	Δ	Δ	Δ	Δ
			DIR	D3	dd	5								
			EXT	F3	hh ll	6								
			IND, X	E3	ff	6								
			IND, Y	18 E3	ff	7								
ANDA (opr)	AND A with memory	A • M ⇒ A	A IMM	84	ii	2	—	—	—	—	Δ	Δ	0	—
			A DIR	94	dd	3								
			A EXT	B4	hh ll	4								
			A IND, X	A4	ff	4								
			A IND, Y	18 A4	ff	5								
ANDB (opr)	AND B with memory	B • M ⇒ B	B IMM	C4	ii	2	—	—	—	—	Δ	Δ	0	—
			B DIR	D4	dd	3								
			B EXT	F4	hh ll	4								
			B IND, X	E4	ff	4								
			B IND, Y	18 E4	ff	5								
ASL (opr)	Arithmetic shift left		EXT	78	hh ll	6	—	—	—	—	Δ	Δ	Δ	Δ
			IND, X	68	ff	6								
			IND, Y	18 68	ff	7								
ASLA	Arithmetic shift left A		A INH	48	—	2	—	—	—	—	Δ	Δ	Δ	Δ
ASLB	Arithmetic shift left B		B INH	58	—	2	—	—	—	—	Δ	Δ	Δ	Δ
ASLD	Arithmetic shift left D		INH	05	—	3	—	—	—	—	Δ	Δ	Δ	Δ
ASR	Arithmetic shift right		EXT	77	hh ll	6	—	—	—	—	Δ	Δ	Δ	Δ
			IND, X	67	ff	6								
			IND, Y	18 67	ff	7								
ASRA	Arithmetic shift right A		A INH	47	—	2	—	—	—	—	Δ	Δ	Δ	Δ
ASRB	Arithmetic shift right B		B INH	57	—	2	—	—	—	—	Δ	Δ	Δ	Δ
BCC (rel)	Branch if carry clear	C = 0 ?	REL	24	rr	3	—	—	—	—	—	—	—	—
BCLR (opr) (msk)	Clear bit(s)	M • ($\overline{MM}$) ⇒ M	DIR	15	dd mm	6	—	—	—	—	Δ	Δ	0	—
			IND, X	1D	ff mm	7								
			IND, Y	18 1D	ff mm	8								
BCS (rel)	Branch if carry set	C = 1 ?	REL	25	rr	3	—	—	—	—	—	—	—	—
BEQ (rel)	Branch if equal to zero	Z = 1 ?	REL	27	rr	3	—	—	—	—	—	—	—	—
BGE (rel)	Branch if ≥ zero	N ⊕ V = 0 ?	REL	2C	rr	3	—	—	—	—	—	—	—	—
BGT (rel)	Branch if > zero	Z + (N ⊕ V) = 0 ?	REL	2E	rr	3	—	—	—	—	—	—	—	—
BHI (rel)	Branch if higher	C + Z = 0 ?	REL	22	rr	3	—	—	—	—	—	—	—	—

Table 3-2 Instruction set (Page 2 of 6)

Mnemonic	Operation	Description	Addressing mode	Instruction			Condition codes							
				Opcode	Operand	Cycles	S	X	H	I	N	Z	V	C
BHS (rel)	Branch if higher or same	C = 0 ?	REL	24	rr	3	—	—	—	—	—	—	—	—
BITA (opr)	Bit(s) test A with memory	A • M	A IMM A DIR A EXT A IND, X A IND, Y	85 95 B5 A5 18 A5	ii dd hh ll ff ff	2 3 4 4 5	—	—	—	—	Δ	Δ	0	—
BITB (opr)	Bit(s) test B with memory	B • M	B IMM B DIR B EXT B IND, X B IND, Y	C5 D5 F5 E5 18 E5	ii dd hh ll ff ff	2 3 4 4 5	—	—	—	—	Δ	Δ	0	—
BLE (rel)	Branch if ≤ zero	Z + (N ⊕ V) = 1 ?	REL	2F	rr	3	—	—	—	—	—	—	—	—
BLO (rel)	Branch if lower	C = 1 ?	REL	25	rr	3	—	—	—	—	—	—	—	—
BLS (rel)	Branch if lower or same	C + Z = 1 ?	REL	23	rr	3	—	—	—	—	—	—	—	—
BLT (rel)	Branch if < zero	N ⊕ V = 1 ?	REL	2D	rr	3	—	—	—	—	—	—	—	—
BMI (rel)	Branch if minus	N = 1 ?	REL	2B	rr	3	—	—	—	—	—	—	—	—
BNE (rel)	Branch if ≠ zero	Z = 0 ?	REL	26	rr	3	—	—	—	—	—	—	—	—
BPL(rel)	Branch if plus	N = 0 ?	REL	2A	rr	3	—	—	—	—	—	—	—	—
BRA (rel)	Branch always	1 = 1 ?	REL	20	rr	3	—	—	—	—	—	—	—	—
BRCLR(opr) (msk) (rel)	Branch if bit(s) clear	M • mm = 0 ?	DIR IND, X IND, Y	13 1F 18 1F	dd mm rr ff mm rr ff mm rr	6 7 8	—	—	—	—	—	—	—	—
BRN (rel)	Branch never	1 = 0 ?	REL	21	rr	3	—	—	—	—	—	—	—	—
BRSET(opr) (msk) (rel)	Branch if bit(s) set	$\overline{M}$ • mm = 0 ?	DIR IND, X IND, Y	12 1E 18 1E	dd mm rr ff mm rr ff mm rr	6 7 8	—	—	—	—	—	—	—	—
BSET (opr) (msk)	Set bit(s)	M + mm ⇒ M	DIR IND, X IND, Y	14 1C 18 1C	dd mm ff mm ff mm	6 7 8	—	—	—	—	Δ	Δ	0	—
BSR (rel)	Branch to subroutine	see Figure 3-2	REL	8D	rr	6	—	—	—	—	—	—	—	—
BVC (rel)	Branch if overflow clear	V = 0 ?	REL	28	rr	3	—	—	—	—	—	—	—	—
BVS (rel)	Branch if overflow set	V = 1 ?	REL	29	rr	3	—	—	—	—	—	—	—	—
CBA	Compare A with B	A – B	INH	11	—	2	—	—	—	—	Δ	Δ	Δ	Δ
CLC	Clear carry bit	0 ⇒ C	INH	0C	—	2	—	—	—	—	—	—	—	0
CLI	Clear interrupt mask	0 ⇒ I	INH	0E	—	2	—	—	—	0	—	—	—	—
CLR (opr)	Clear memory byte	0 ⇒ M	DIR IND, X IND, Y	7F 6F 18 6F	hh ll ff ff	6 6 7	—	—	—	—	0	1	0	0
CLRA	Clear accumulator A	0 ⇒ A	A INH	4F	—	2	—	—	—	—	0	1	0	0
CLRB	Clear accumulator B	0 ⇒ B	B INH	5F	—	2	—	—	—	—	0	1	0	0
CLV	Clear overflow flag	0 ⇒ V	INH	0A	—	2	—	—	—	—	—	—	0	—
CMPA (opr)	Compare A with memory	A – M	A IMM A DIR A EXT A IND, X A IND, Y	81 91 B1 A1 18 A1	ii dd hh ll ff ff	2 3 4 4 5	—	—	—	—	Δ	Δ	Δ	Δ
CMPB (opr)	Compare B with memory	B – M	B IMM B DIR B EXT B IND, X B IND, Y	C1 D1 F1 E1 18 E1	ii dd hh ll ff ff	2 3 4 4 5	—	—	—	—	Δ	Δ	Δ	Δ
COM (opr)	Ones complement memory byte	$FF – M ⇒ M	EXT IND, X IND, Y	73 63 18 63	hh ll ff ff	6 6 7	—	—	—	—	Δ	Δ	0	1
COMA	Ones complement A	$FF – A ⇒ A	A INH	43	—	2	—	—	—	—	Δ	Δ	0	1
COMB	Ones complement B	$FF – B ⇒ B	B INH	53	—	2	—	—	—	—	Δ	Δ	0	1

Table 3-2 Instruction set (Page 3 of 6)

Mnemonic	Operation	Description	Addressing mode	Instruction			Condition codes							
				Opcode	Operand	Cycles	S	X	H	I	N	Z	V	C
CPD (opr)	Compare D with memory (16-bit)	D – (M:M+1)	IMM DIR EXT IND, X IND, Y	1A 83 1A 93 1A B3 1A A3 CD A3	jj kk dd hh ll ff ff	5 6 7 7 7	—	—	—	—	Δ	Δ	Δ	Δ
CPX (opr)	Compare IX with memory (16-bit)	IX – (M:M+1)	IMM DIR EXT IND, X IND, Y	8C 9C BC AC CD AC	jj kk dd hh ll ff ff	4 5 6 6 7	—	—	—	—	Δ	Δ	Δ	Δ
CPY (opr)	Compare IY with memory (16-bit)	IY – (M:M+1)	IMM DIR EXT IND, X IND, Y	18 8C 18 9C 18 BC 1A AC 18 AC	jj kk dd hh ll ff ff	5 6 7 7 7	—	—	—	—	Δ	Δ	Δ	Δ
DAA	Decimal adjust A	adjust sum to BCD	INH	19	—	2	—	—	—	—	Δ	Δ	?	Δ
DEC (opr)	Decrement memory byte	M – 1 ⇒ M	EXT IND, X IND, Y	7A 6A 18 6A	hh ll ff ff	6 6 7	—	—	—	—	Δ	Δ	Δ	—
DECA	Decrement accumulator A	A – 1 ⇒ A	A INH	4A	—	2	—	—	—	—	Δ	Δ	Δ	—
DECB	Decrement accumulator B	B – 1 ⇒ B	B INH	5A	—	2	—	—	—	—	Δ	Δ	Δ	—
DES	Decrement stack pointer	SP – 1 ⇒ SP	INH	34	—	3	—	—	—	—	—	—	—	—
DEX	Decrement index register X	IX – 1 ⇒ IX	INH	09	—	3	—	—	—	—	—	Δ	—	—
DEY	Decrement index register Y	IY – 1 ⇒ IY	INH	18 09	—	4	—	—	—	—	—	Δ	—	—
EORA (opr)	Exclusive OR A with memory	A ⊕ M ⇒ A	A IMM A DIR A EXT A IND, X A IND, Y	88 98 B8 A8 18 A8	ii dd hh ll ff ff	2 3 4 4 5	—	—	—	—	Δ	Δ	0	—
EORB (opr)	Exclusive OR B with memory	B ⊕ M ⇒ A	B IMM B DIR B EXT B IND, X B IND, Y	C8 D8 F8 E8 18 E8	ii dd hh ll ff ff	2 3 4 4 5	—	—	—	—	Δ	Δ	0	—
FDIV	Fractional divide, 16 by 16	D / IX ⇒ IX; r ⇒ D	INH	03	—	41	—	—	—	—	—	Δ	Δ	Δ
IDIV	Integer divide, 16 by 16	D / IX ⇒ IX; r ⇒ D	INH	02	—	41	—	—	—	—	—	Δ	0	Δ
INC (opr)	Increment memory byte	M + 1 ⇒ M	EXT IND, X IND, Y	7C 6C 18 6C	hh ll ff ff	6 6 7	—	—	—	—	Δ	Δ	Δ	—
INCA	Increment accumulator A	A + 1 ⇒ A	A INH	4C	—	2	—	—	—	—	Δ	Δ	Δ	—
INCB	Increment accumulator B	B + 1 ⇒ B	B INH	5C	—	2	—	—	—	—	Δ	Δ	Δ	—
INS	Increment stack pointer	SP + 1 ⇒ SP	INH	31	—	3	—	—	—	—	—	—	—	—
INX	Increment index register X	IX + 1 ⇒ IX	INH	08	—	3	—	—	—	—	—	Δ	—	—
INY	Increment index register Y	IY + 1 ⇒ IY	INH	18 08	—	4	—	—	—	—	—	Δ	—	—
JMP (opr)	Jump	see Figure 3-2	EXT IND, X IND, Y	7E 6E 18 6E	hh ll ff ff	3 3 4	—	—	—	—	—	—	—	—
JSR (opr)	Jump to subroutine	see Figure 3-2	DIR EXT IND, X IND, Y	9D BD AD 18 AD	dd hh ll ff ff	5 6 6 7	—	—	—	—	—	—	—	—
LDAA (opr)	Load accumulator A	M ⇒ A	A IMM A DIR A EXT A IND, X A IND, Y	86 96 B6 A6 18 A6	ii dd hh ll ff ff	2 3 4 4 5	—	—	—	—	Δ	Δ	0	—

Table 3-2 Instruction set (Page 4 of 6)

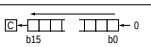
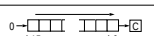
Mnemonic	Operation	Description	Addressing mode	Instruction			Condition codes							
				Opcode	Operand	Cycles	S	X	H	I	N	Z	V	C
LDAB (opr)	Load accumulator B	M ⇒ B	B IMM B DIR B EXT B IND, X B IND, Y	C6 D6 F6 E6 18 E6	ii dd hh ll ff ff	2 3 4 4 5	—	—	—	—	Δ	Δ	0	—
LDD (opr)	Load double accumulator D	M ⇒ A; M+1 ⇒ B	IMM DIR EXT IND, X IND, Y	CC DC FC EC 18 EC	jj kk dd hh ll ff ff	3 4 5 5 6	—	—	—	—	Δ	Δ	0	—
LDS (opr)	Load stack pointer	M:M+1 ⇒ SP	IMM DIR EXT IND, X IND, Y	8E 9E BE AE 18 AE	jj kk dd hh ll ff ff	3 4 5 5 6	—	—	—	—	Δ	Δ	0	—
LDX (opr)	Load index register X	M:M+1 ⇒ IX	IMM DIR EXT IND, X IND, Y	CE DE FE EE CD EE	jj kk dd hh ll ff ff	3 4 5 5 6	—	—	—	—	Δ	Δ	0	—
LDY (opr)	Load index register Y	M:M+1 ⇒ IY	IMM DIR EXT IND, X IND, Y	18 CE 18 DE 18 FE 1A EE 18 EE	jj kk dd hh ll ff ff	4 5 6 6 6	—	—	—	—	Δ	Δ	0	—
LSL (opr)	Logical shift left		EXT IND, X IND, Y	78 68 18 68	hh ll ff ff	6 6 7	—	—	—	—	Δ	Δ	Δ	Δ
LSLA	Logical shift left A		A INH	48	—	2	—	—	—	—	Δ	Δ	Δ	Δ
LSLB	Logical shift Left B		B INH	58	—	2	—	—	—	—	Δ	Δ	Δ	Δ
LSLD	Logical shift left D		INH	05	—	3	—	—	—	—	Δ	Δ	Δ	Δ
LSR (opr)	Logical shift right		EXT IND, X IND, Y	74 64 18 64	hh ll ff ff	6 6 7	—	—	—	—	0	Δ	Δ	Δ
LSRA	Logical shift right A		A INH	44	—	2	—	—	—	—	0	Δ	Δ	Δ
LSRB	Logical shift right B		B INH	54	—	2	—	—	—	—	0	Δ	Δ	Δ
LSRD	Logical shift right D		INH	04	—	3	—	—	—	—	0	Δ	Δ	Δ
MUL	Multiply, 8 x 8	A * B ⇒ D	INH	3D	—	10	—	—	—	—	—	—	—	Δ
NEG (opr)	Twos complement memory byte	0 – M ⇒ M	EXT IND, X IND, Y	70 60 18 60	hh ll ff ff	6 6 7	—	—	—	—	Δ	Δ	Δ	Δ
NEGA	Twos complement A	0 – A ⇒ A	A INH	40	—	2	—	—	—	—	Δ	Δ	Δ	Δ
NEGB	Twos complement B	0 – B ⇒ B	B INH	50	—	2	—	—	—	—	Δ	Δ	Δ	Δ
NOP	No operation	no operation	INH	01	—	2	—	—	—	—	—	—	—	—
ORAA	OR accumulator A (inclusive)	A + M ⇒ A	A IMM A DIR A EXT A IND, X A IND, Y	8A 9A BA AA 18 AA	ii dd hh ll ff ff	2 3 4 4 5	—	—	—	—	Δ	Δ	0	—
ORAB	OR accumulator B (inclusive)	B + M ⇒ B	B IMM B DIR B EXT B IND, X B IND, Y	CA DA FA EA 18 EA	ii dd hh ll ff ff	2 3 4 4 5	—	—	—	—	Δ	Δ	0	—
PSHA	Push A onto stack	A ⇒ Stack; SP = SP–1	A INH	36	—	3	—	—	—	—	—	—	—	—
PSHB	Push B onto stack	B ⇒ Stack; SP = SP–1	B INH	37	—	3	—	—	—	—	—	—	—	—
PSHX	Push IX onto stack (low first)	IX ⇒ Stack; SP = SP–2	INH	3C	—	4	—	—	—	—	—	—	—	—
PSHY	Push IY onto stack (low first)	IY ⇒ Stack; SP = SP–2	INH	18 3C	—	5	—	—	—	—	—	—	—	—

Table 3-2 Instruction set (Page 5 of 6)

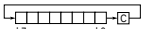
Mnemonic	Operation	Description	Addressing mode	Instruction			Condition codes							
				Opcode	Operand	Cycles	S	X	H	I	N	Z	V	C
PULA	Pull A from stack	SP = SP+1; Stack ⇒ A	A INH	32	—	4	—	—	—	—	—	—	—	—
PULB	Pull B from stack	SP = SP+1; Stack ⇒ B	B INH	33	—	4	—	—	—	—	—	—	—	—
PULX	Pull IX from stack (high first)	SP = SP+2; Stack ⇒ IX	INH	38	—	5	—	—	—	—	—	—	—	—
PULY	Pull IY from stack (high first)	SP = SP+2; Stack ⇒ IY	INH	18 38	—	6	—	—	—	—	—	—	—	—
ROL (opr)	Rotate left		EXT IND, X IND, Y	79 69 18 69	hh ll ff ff	6 6 7	—	—	—	—	Δ	Δ	Δ	Δ
ROLA	Rotate left A		A INH	49	—	2	—	—	—	—	Δ	Δ	Δ	Δ
ROLB	Rotate left B		B INH	59	—	2	—	—	—	—	Δ	Δ	Δ	Δ
ROR (opr)	Rotate right		EXT IND, X IND, Y	76 66 18 66	hh ll ff ff	6 6 7	—	—	—	—	Δ	Δ	Δ	Δ
RORA	Rotate right A		A INH	46	—	2	—	—	—	—	Δ	Δ	Δ	Δ
RORB	Rotate right B		B INH	56	—	2	—	—	—	—	Δ	Δ	Δ	Δ
RTI	Return from interrupt	see Figure 3-2	INH	3B	—	12	Δ	↓	Δ	Δ	Δ	Δ	Δ	Δ
RTS	Return from subroutine	see Figure 3-2	INH	39	—	5	—	—	—	—	—	—	—	—
SBA	Subtract B from A	A – B ⇒ A	INH	10	—	2	—	—	—	—	Δ	Δ	Δ	Δ
SBCA (opr)	Subtract with carry from A	A – M – C ⇒ A	A IMM A DIR A EXT A IND, X A IND, Y	82 92 B2 A2 18 A2	ii dd hh ll ff ff	2 3 4 4 5	—	—	—	—	Δ	Δ	Δ	Δ
SBCB (opr)	Subtract with carry from B	B – M – C ⇒ B	B IMM B DIR B EXT B IND, X B IND, Y	C2 D2 F2 E2 18 E2	ii dd hh ll ff ff	2 3 4 4 5	—	—	—	—	Δ	Δ	Δ	Δ
SEC	Set carry	1 ⇒ C	INH	0D	—	2	—	—	—	—	—	—	—	1
SEI	Set interrupt mask	1 ⇒ I	INH	0F	—	2	—	—	—	1	—	—	—	—
SEV	Set overflow flag	1 ⇒ V	INH	0B	—	2	—	—	—	—	—	—	1	—
STAA (opr)	Store accumulator A	A ⇒ M	A DIR A EXT A IND, X A IND, Y	97 B7 A7 18 A7	dd hh ll ff ff	3 4 4 5	—	—	—	—	Δ	Δ	0	—
STAB (opr)	Store accumulator B	B ⇒ M	B DIR B EXT B IND, X B IND, Y	D7 F7 E7 18 E7	dd hh ll ff ff	3 4 4 5	—	—	—	—	Δ	Δ	0	—
STD (opr)	Store accumulator D	A ⇒ M; B ⇒ M+1	DIR EXT IND, X IND, Y	DD FD ED 18 ED	dd hh ll ff ff	4 5 5 6	—	—	—	—	Δ	Δ	0	—
STOP	Stop internal clocks	—	INH	CF	—	2	—	—	—	—	—	—	—	—
STS (opr)	Store stack pointer	SP ⇒ M:M+1	DIR EXT IND, X IND, Y	9F BF AF 18 AF	dd hh ll ff ff	4 5 5 6	—	—	—	—	Δ	Δ	0	—
STX (opr)	Store index register X	IX ⇒ M:M+1	DIR EXT IND, X IND, Y	DF FF EF CD EF	dd hh ll ff ff	4 5 5 6	—	—	—	—	Δ	Δ	0	—
STY (opr)	Store index register Y	IY ⇒ M:M+1	DIR EXT IND, X IND, Y	18 DF 18 FF 1A EF 18 EF	dd hh ll ff ff	5 6 6 6	—	—	—	—	Δ	Δ	0	—

Table 3-2 Instruction set (Page 6 of 6)

Mnemonic	Operation	Description	Addressing mode	Instruction			Condition codes							
				Opcode	Operand	Cycles	S	X	H	I	N	Z	V	C
SUBA (opr)	Subtract memory from A	A – M ⇒ A	A IMM A DIR A EXT A IND, X A IND, Y	80 90 B0 A0 18 A0	ii dd hh ll ff ff	2 3 4 4 5	—	—	—	—	Δ	Δ	Δ	Δ
SUBB (opr)	Subtract memory from B	B – M ⇒ B	B IMM B DIR B EXT B IND, X B IND, Y	C0 D0 F0 E0 18 E0	ii dd hh ll ff ff	2 3 4 4 5	—	—	—	—	Δ	Δ	Δ	Δ
SUBD (opr)	Subtract memory from D	D – M:M+1 ⇒ D	IMM DIR EXT IND, X IND, Y	83 93 B3 A3 18 A3	jj kk dd hh ll ff ff	4 5 6 6 7	—	—	—	—	Δ	Δ	Δ	Δ
SWI	Software interrupt	see Figure 3-2	INH	3F	—	14	—	—	—	1	—	—	—	—
TAB	Transfer A to B	A ⇒ B	INH	16	—	2	—	—	—	—	Δ	Δ	0	—
TAP	Transfer A to CC register	A ⇒ CCR	INH	06	—	2	Δ	↓	Δ	Δ	Δ	Δ	Δ	Δ
TBA	Transfer B to A	B ⇒ A	INH	17	—	2	—	—	—	—	Δ	Δ	0	—
TEST	Test (only in test modes)	address bus increments	INH	00	—	†	—	—	—	—	—	—	—	—
TPA	Transfer CC register to A	CCR ⇒ A	INH	07	—	2	—	—	—	—	—	—	—	—
TST (opr)	Test for zero or minus	M – 0	EXT IND, X IND, Y	7D 6D 18 6D	hh ll ff ff	6 6 7	—	—	—	—	Δ	Δ	0	0
TSTA	Test A for zero or minus	A – 0	A INH	4D	—	2	—	—	—	—	Δ	Δ	0	0
TSTB	Test B for zero or minus	B – 0	B INH	5D	—	2	—	—	—	—	Δ	Δ	0	0
TSX	Transfer stack pointer to X	SP + 1 ⇒ IX	INH	30	—	3	—	—	—	—	—	—	—	—
TSY	Transfer stack pointer to Y	SP + 1 ⇒ IY	INH	18 30	—	4	—	—	—	—	—	—	—	—
TXS	Transfer X to stack pointer	IX – 1 ⇒ SP	INH	35	—	3	—	—	—	—	—	—	—	—
TYS	Transfer Y to stack pointer	IY – 1 ⇒ SP	INH	18 35	—	4	—	—	—	—	—	—	—	—
WAI	Wait for interrupt	stack registers & WAIT	INH	3E	—	‡	—	—	—	—	—	—	—	—
XGDX	Exchange D with X	IX ⇒ D; D ⇒ IX	INH	8F	—	3	—	—	—	—	—	—	—	—
XGDY	Exchange D with Y	IY ⇒ D; D ⇒ IY	INH	18 8F	—	4	—	—	—	—	—	—	—	—

Operators

- ⇒ Is transferred to
- • Boolean AND
- \+ Arithmetic addition, except where used as an inclusive-OR symbol in Boolean formulae
- ⊕ Exclusive-OR
- * Multiply
- : Concatenation
- – Arithmetic subtraction, or negation symbol (Twos complement)

Operands

- dd 8-bit direct address ($0000–$00FF); the high byte is assumed to be zero
- ff 8-bit positive offset ($00 to $FF (0 to 256)) is added to the contents of the index register
- hh High order byte of 16-bit extended address
- ii One byte of immediate data
- jj High order byte of 16-bit immediate data
- kk Low order byte of 16-bit immediate data
- ll Low order byte of 16-bit extended address
- mm 8-bit mask (set bits to be affected)
- rr Signed relative offset ($80 to $7F (–128 to +127)); offset is relative to the address following the offset byte

Cycles

- † Infinite, or until reset occurs
- ‡ 12 cycles are used, beginning with the opcode fetch. A wait state is entered, which remains in effect for an integer number of MPU E clock cycles (n) until an interrupt is recognised. Finally, two additional cycles are used to fetch the appropriate interrupt vector. (14 + n, total).

Condition Codes

- — Bit not changed
- 0 Bit always cleared
- 1 Bit always set
- Δ Bit set or cleared, depending on the operation
- ↓ Bit can be cleared, but cannot become set
- ? Not defined

4
OPERATING MODES AND ON-CHIP MEMORY

This section contains information about the modes that define MC68HC11KW1 operating conditions, and about the on-chip memory that allows the MCU to be configured for various applications.

4.1 Operating modes

The values of the mode select inputs MODB and MODA during reset determine the operating mode (See Table 4-4). Single chip and expanded modes are the normal modes. In single chip mode only on-board memory is available. Expanded mode, however, allows access to external memory. Each of these two normal modes is paired with a special mode. Bootstrap, a variation of the single chip mode, is a special mode that executes a bootloader program in an internal bootstrap ROM. Test is a special expanded mode that allows privileged access to internal resources.

4.1.1 Single chip operating mode

In single chip operating mode, the MC68HC11KW1 microcontroller has no external address or data bus. Ports B, C and F are available for general-purpose parallel I/O.

4.1.2 Expanded operating mode

In expanded operating mode, the MCU can access a 64K byte physical address space. The address space includes the same on-chip memory addresses used for single chip mode, in addition to external memory and peripheral devices.The expansion bus is made up of ports B, C, and F. In expanded mode, high order address bits are output on the port B pins, low order address bits on the port F pins, and the data bus on port C. The R/$\overline{W}$ pin signals the direction of data transfer on the port C bus.

The MC68HC11KW1 includes an additional memory expansion feature, available in expanded modes, which allows access to pages of memory in one or two windows within the 64K byte physical memory space. This can extend the memory space to more than 1M byte. See Section 4.4.

For information about the four programmable chip selects available in expanded modes, refer to Section 4.5. A security feature can protect EEPROM data when in expanded mode (see Section 4.6.3).

4.1.3 Special test mode

Special test, a variation of the expanded mode, is used during Motorola's internal production testing, and is not intended or recommended for any other purpose. Its specification is subject to change without notice.

4.1.4 Special bootstrap mode

When the MCU is reset in special bootstrap mode, a small on-chip ROM is enabled at address $BE40–$BFFF. The ROM contains a reset vector and a bootloader program. The MCU fetches the reset vector, then executes the bootloader.

For normal use of the bootloader program, send a synchronization byte $FF to the SCI receiver at either E clock ÷256, or E clock ÷1664 (15624 or 2400 baud respectively, for an E clock of 4 MHz). Then download up to 768 bytes of program data (which is put into RAM starting at $00A0). These characters are echoed through the transmitter. The bootloader program ends the download after a timeout of four character times or 768 bytes. When loading is complete, the program jumps to location $00A0 and begins executing the code. Use of an external pull-up resistor is required when using the SCI transmitter pin (TXD) because port D pins are configured for wired-OR operation by the bootloader. In bootstrap mode, the interrupt vectors point to RAM. This allows the use of interrupts through a jump table.

Further baud rate options are available on the MC68HC11KW1 by using a different value for the synchronization byte, as shown in Table 4-1.

Refer also to Motorola application note ***AN1060, M68HC11 Bootstrap Mode*** (the bootloader mode is similar to that used on the MC68HC11K4).

Table 4-1 Example bootloader baud rates

Sync. byte	Timeout delay	Baud rates for an E clock of 4.00 MHz
$FF	4 char.	15624
$FF	4	2400
$F0	4.9	19200
$FD	17.3	10416
$FD	13	7812

4.2 On-chip memory

The MC68HC11KW1 MCU includes 768 bytes of on-chip RAM and 640 bytes of EEPROM. The bootloader ROM occupies a 448 byte block of the memory map. The CONFIG register is implemented as a separate EEPROM byte.

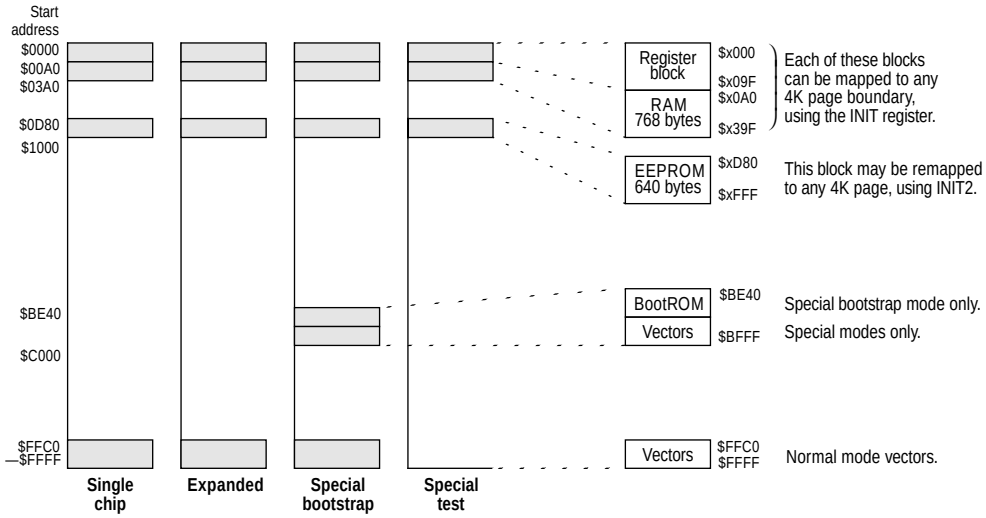


Figure 4-1 MC68HC11KW1 memory map

4.2.1 Mapping allocations

Memory locations for on-chip resources are the same for both expanded and single chip modes. The 160-byte register block originates at $0000 on reset and can be placed at any other 4K boundary ($x000) after reset by writing an appropriate value to the INIT register. Refer to Figure 4-1, which shows the memory map.

The on-board 768 byte block of RAM is initially located at $00A0 after reset. The RAM is divided into two sections of 160 bytes and 608 bytes. If RAM and registers are both mapped to the same 4K boundary, RAM starts at $x0A0 and 160 bytes are remapped at $x300–$x39F. Otherwise, RAM starts at $x000. See Figure 4-2.

Remapping is accomplished by writing appropriate values into the two nibbles of the INIT register. See Section 4.3.2.2.

The 640-byte EEPROM is initially located at $0D80 after reset, when EEPROM is enabled in the memory map by the CONFIG register. EEPROM can be placed in any other 4K page ($xD80) by writing to the INIT2 register.

In special bootstrap mode, a bootloader ROM is enabled at locations $BE40–$BFFF. The vectors for special bootstrap mode are contained in the bootloader program.

4.2.1.1 RAM

The MC68HC11KW1 has 768 bytes of fully static RAM that are used for storing instructions, variables and temporary data during program execution. RAM can be placed at any 4K boundary in the 64K byte address space by writing an appropriate value to the INIT register.

By default, RAM is initially located at $00A0 in the memory map. Direct addressing mode can access the first 96 locations of RAM using a one-byte address operand. Direct mode accesses save program memory space and execution time. Registers can be moved to other boundaries to allow 256 bytes of RAM to be located in direct addressing space. See Figure 4-2.

The on-chip RAM is a fully static memory. RAM contents can be preserved during periods of processor inactivity by either of two methods, both of which reduce power consumption:

1) During the software-based STOP mode, MCU clocks are stopped, but the MCU continues to draw power from V_{DD}. Power supply current is directly related to operating frequency in CMOS integrated circuits and there is very little leakage when the clocks are stopped. These two factors reduce power consumption while the MCU is in STOP mode.
2) To reduce power consumption to a minimum, V_{DD} can be turned off, and the MODB/VSTBY pin can be used to supply RAM power from either a battery back-up or a second power supply. Although this method requires external hardware, it is very effective. Refer to Section 2 for information about how to connect the stand-by RAM power supply and to Section 5 for a description of low power operation.

4.2.1.2 Bootloader ROM

The bootloader ROM is enabled at address $BE40–$BFFF during special bootstrap mode. The reset vector is fetched from this ROM and the MCU executes the bootloader firmware. In normal modes, the bootloader ROM is disabled.

4.2.2 Registers

In Table 4-2, a summary of registers and control bits, the registers are shown in ascending order within the 160-byte register block. The addresses shown are for default block mapping ($0000–$009F), however the INIT register remaps the block to any 4K page ($x000–$x09F). See Section 4.3.2.2.

Table 4-2 Register and control bit assignments (Page 1 of 5)

Register name	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Port A data (PORTA)	$0000	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0	undefined
Data direction A (DDRA)	$0001	DDA7	DDA6	DDA5	DDA4	DDA3	DDA2	DDA1	DDA0	0000 0000
Data direction B (DDRB)	$0002	DDB7	DDB6	DDB5	DDB4	DDB3	DDB2	DDB1	DDB0	0000 0000
Data direction F (DDRF)	$0003	DDF7	DDF6	DDF5	DDF4	DDF3	DDF2	DDF1	DDF0	0000 0000
Port B data (PORTB)	$0004	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0	undefined
Port F data (PORTF)	$0005	PF7	PF6	PF5	PF4	PF3	PF2	PF1	PF0	undefined
Port C data (PORTC)	$0006	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0	undefined
Data direction C (DDRC)	$0007	DDC7	DDC6	DDC5	DDC4	DDC3	DDC2	DDC1	DDC0	0000 0000
Port D data (PORTD)	$0008	PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0	undefined
Data direction D (DDRD)	$0009	DDD7	DDD6	DDD5	DDD4	DDD3	DDD2	DDD1	DDD0	0000 0000
Port E data (PORTE)	$000A	PE7	PE6	PE5	PE4	PE3	PE2	PE1	PE0	undefined
Timer compare force (CFORC)	$000B	FOC1	FOC2	FOC3	FOC4	FOC5	0	0	0	0000 0000
Output compare 1 mask (OC1M)	$000C	OC1M7	OC1M6	OC1M5	OC1M4	OC1M3	0	0	0	0000 0000
Output compare 1 data (OC1D)	$000D	OC1D7	OC1D6	OC1D5	OC1D4	OC1D3	0	0	0	0000 0000
Timer count (TCNT) high	$000E	(bit 15)	(14)	(13)	(12)	(11)	(10)	(9)	(bit 8)	0000 0000
Timer count (TCNT) low	$000F	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	0000 0000
Timer input capture 1 (TIC1) high	$0010	(bit 15)	(14)	(13)	(12)	(11)	(10)	(9)	(bit 8)	undefined
Timer input capture 1 (TIC1) low	$0011	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	undefined
Timer input capture 2 (TIC2) high	$0012	(bit 15)	(14)	(13)	(12)	(11)	(10)	(9)	(bit 8)	undefined
Timer input capture 2 (TIC2) low	$0013	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	undefined
Timer input capture 3 (TIC3) high	$0014	(bit 15)	(14)	(13)	(12)	(11)	(10)	(9)	(bit 8)	undefined
Timer input capture 3 (TIC3) low	$0015	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	undefined
Timer output compare 1 (TOC1) high	$0016	(bit 15)	(14)	(13)	(12)	(11)	(10)	(9)	(bit 8)	1111 1111
Timer output compare 1 (TOC1) low	$0017	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	1111 1111
Timer output compare 2 (TOC2) high	$0018	(bit 15)	(14)	(13)	(12)	(11)	(10)	(9)	(bit 8)	1111 1111
Timer output compare 2 (TOC2) low	$0019	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	1111 1111
Timer output compare 3 (TOC3) high	$001A	(bit 15)	(14)	(13)	(12)	(11)	(10)	(9)	(bit 8)	1111 1111
Timer output compare 3 (TOC3) low	$001B	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	1111 1111
Timer output compare 4 (TOC4) high	$001C	(bit 15)	(14)	(13)	(12)	(11)	(10)	(9)	(bit 8)	1111 1111
Timer output compare 4 (TOC4) low	$001D	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	1111 1111
Capture 4/compare 5 (TI4/O5) high	$001E	(bit 15)	(14)	(13)	(12)	(11)	(10)	(9)	(bit 8)	1111 1111
Capture 4/compare 5 (TI4/O5) low	$001F	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	1111 1111
Timer control 1 (TCTL1)	$0020	OM2	OL2	OM3	OL3	OM4	OL4	OM5	OL5	0000 0000
Timer control 2 (TCTL2)	$0021	EDG4B	EDG4A	EDG1B	EDG1A	EDG2B	EDG2A	EDG3B	EDG3A	0000 0000
Timer interrupt mask 1 (TMSK1)	$0022	OC1I	OC2I	OC3I	OC4I	I4/O5I	IC1I	IC2I	IC3I	0000 0000

Table 4-2 Register and control bit assignments (Page 2 of 5)

Register name	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Timer interrupt flag 1 (TFLG1)	$0023	OC1F	OC2F	OC3F	OC4F	I4/O5F	IC1F	IC2F	IC3F	0000 0000
Timer interrupt mask 2 (TMSK2)	$0024	TOI	RTII	PAOVI	PAII	0	0	PR1	PR0	0000 0000
Timer interrupt flag 2 (TFLG2)	$0025	TOF	RTIF	PAOVF	PAIF	0	0	0	0	0000 0000
Pulse accumulator control (PACTL)	$0026	0	PAEN	PAMOD	PEDGE	0	I4/O5	RTR1	RTR0	0000 0000
Pulse accumulator count (PACNT)	$0027	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	undefined
SPI control (SPCR)	$0028	SPIE	SPE	DWOM	MSTR	CPOL	CPHA	SPR1	SPR0	0000 01uu
SPI status (SPSR)	$0029	SPIF	WCOL	0	MODF	0	0	0	0	0000 0000
SPI data (SPDR)	$002A	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	undefined
Reserved	$002B									
Port pull-up assignment (PPAR)	$002C	0	0	0	0	HPPUE	GPPUE	FPPUE	BPPUE	0000 1111
Port G assignment (PGAR)	$002D	0	0	PGAR5	PGAR4	PGAR3	PGAR2	PGAR1	PGAR0	0000 0000
Reserved	$002E									
Reserved	$002F									
A/D control & status (ADCTL)	$0030	CCF	CONV8	SCAN	MULT	CD	CC	CB	CA	0000 0000
Compare force for timers 2 and 3 (F23FRC)	$0031	FT3C1	FT3C2	FT3C3	FT3C4	FT2C1	FT2C2	FT2C3	FT2C4	0000 0000
A/D frequency select (ADFRQ)	$0032	0	0	0	0	0	0	0	ADER	0000 0000
Reserved	$0033									
Reserved	$0034									
Block protect (BPROT)	$0035	BULKP	BIT6	BPRT4	PTCON	BPRT3	BPRT2	BPRT1	BPRT0	1111 1111
Reserved	$0036									
EEPROM mapping (INIT2)	$0037	EE3	EE2	EE1	EE0	0	0	0	0	0000 0000
System config. options 2 (OPT2)	$0038	LIRDV	CWOM	0	IRVNE	LSBF	SPR2	XDV1	XDV0	000x 0000
System config. options 1 (OPTION)	$0039	ADPU	CSEL	IRQE	DLY	CME	FCME	CR1	CR0	0001 0000
COP timer arm/reset (COPRST)	$003A	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	undefined
EEPROM programming (PPROG)	$003B	ODD	EVEN	0	BYTE	ROW	ERASE	EELAT	EEPGM	0000 0000
Highest priority interrupt (HPRIO)	$003C	RBOOT	SMOD	MDA	PSEL4	PSEL3	PSEL2	PSEL1	PSEL0	xxx0 0110
RAM & I/O mapping (INIT)	$003D	RAM3	RAM2	RAM1	RAM0	REG3	REG2	REG1	REG0	0000 0000
Factory test (TEST1)	$003E	TILOP	0	OCCR	CBYP	DISR	FCM	FCOP	0	0000 x000
Configuration control (CONFIG)	$003F	1	1	CLKX	PAREN	NOSEC	NOCOP	1	EEON	11xx xx1x
A/D result 1 (ADR1) high	$0040	(Bit 15)	(14)	(13)	(12)	(11)	(10)	(9)	(8)	undefined
A/D result 1 (ADR1) low	$0041	(7)	(6)	0	0	0	0	0	0	uu00 0000
A/D result 2 (ADR2) high	$0042	(Bit 15)	(14)	(13)	(12)	(11)	(10)	(9)	(8)	undefined
A/D result 2 (ADR2) low	$0043	(7)	(6)	0	0	0	0	0	0	uu00 0000

Table 4-2 Register and control bit assignments (Page 3 of 5)

Register name	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
A/D result 3 (ADR3) high	$0044	(Bit 15)	(14)	(13)	(12)	(11)	(10)	(9)	(8)	undefined
A/D result 3 (ADR3) low	$0045	(7)	(6)	0	0	0	0	0	0	uu00 0000
A/D result 4 (ADR4) high	$0046	(Bit 15)	(14)	(13)	(12)	(11)	(10)	(9)	(8)	undefined
A/D result 4 (ADR4) low	$0047	(7)	(6)	0	0	0	0	0	0	uu00 0000
A/D result 5 (ADR5) high	$0048	(Bit 15)	(14)	(13)	(12)	(11)	(10)	(9)	(8)	undefined
A/D result 5 (ADR5) low	$0049	(7)	(6)	0	0	0	0	0	0	uu00 0000
A/D result 6 (ADR6) high	$004A	(Bit 15)	(14)	(13)	(12)	(11)	(10)	(9)	(8)	undefined
A/D result 6 (ADR6) low	$004B	(7)	(6)	0	0	0	0	0	0	uu00 0000
A/D result 7 (ADR7) high	$004C	(Bit 15)	(14)	(13)	(12)	(11)	(10)	(9)	(8)	undefined
A/D result 7 (ADR7) low	$004D	(7)	(6)	0	0	0	0	0	0	uu00 0000
A/D result 8 (ADR8) high	$004E	(Bit 15)	(14)	(13)	(12)	(11)	(10)	(9)	(8)	undefined
A/D result 8 (ADR8) low	$004F	(7)	(6)	0	0	0	0	0	0	uu00 0000
Reserved	$0050									
Reserved	$0051									
Reserved	$0052									
Reserved	$0053									
Reserved	$0054									
Reserved	$0055									
Memory mapping window size (MMSIZ)	$0056	MXGS2	MXGS1	W2SZ1	W2SZ0	0	0	W1SZ1	W1SZ0	0000 0000
Memory mapping window base (MMWBR)	$0057	W2A15	W2A14	W2A13	0	W1A15	W1A14	W1A13	0	0000 0000
Memory mapping window 1 control (MM1CR)	$0058	0	X1A18	X1A17	X1A16	X1A15	X1A14	X1A13	0	0000 0000
Memory mapping window 2 control (MM2CR)	$0059	0	X2A18	X2A17	X2A16	X2A15	X2A14	X2A13	0	0000 0000
Chip select clock stretch (CSCSTR)	$005A	IOSA	IOSB	GP1SA	GP1SB	GP2SA	GP2SB	PCSA	PCSB	0000 000x
Chip select control (CSCTL)	$005B	IOEN	IOPL	IOCSA	IOSZ	GCSPR	PCSEN	PCSZA	PCSZB	0000 0100
Gen. purpose chip select 1 addr. (GPCS1A)	$005C	G1A18	G1A17	G1A16	G1A15	G1A14	G1A13	G1A12	G1A11	0000 0000
Gen. purpose chip select 1 con. (GPCS1C)	$005D	G1DG2	G1DPC	G1POL	G1AV	G1SZA	G1SZB	G1SZC	G1SZD	0000 0000
Gen. purpose chip select 2 addr. (GPCS2A)	$005E	G2A18	G2A17	G2A16	G2A15	G2A14	G2A13	G2A12	G2A11	0000 0000
Gen. purpose chip select 2 con. (GPCS2C)	$005F	0	G2DPC	G2POL	G2AV	G2SZA	G2SZB	G2SZC	G2SZD	0000 0000
Pulse width clock select (PWCLK)	$0060	CON34	CON12	PCKA2	PCKA1	0	PCKB3	PCKB2	PCKB1	0000 0000
Pulse width polarity select (PWPOL)	$0061	PCLK4	PCLK3	PCLK2	PCLK1	PPOL4	PPOL3	PPOL2	PPOL1	0000 0000
Pulse width scale (PWSCAL)	$0062	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	0000 0000
Pulse width enable (PWEN)	$0063	TPWSL	DISCP	0	0	PWEN4	PWEN3	PWEN2	PWEN1	0000 0000
Pulse width count 1 (PWCNT1)	$0064	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	0000 0000

Table 4-2 Register and control bit assignments (Page 4 of 5)

Register name	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Pulse width count 2 (PWCNT2)	$0065	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	0000 0000
Pulse width count 3 (PWCNT3)	$0066	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	0000 0000
Pulse width count 4 (PWCNT4)	$0067	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	0000 0000
Pulse width period 1 (PWPER1)	$0068	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	1111 1111
Pulse width period 2 (PWPER2)	$0069	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	1111 1111
Pulse width period 3 (PWPER3)	$006A	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	1111 1111
Pulse width period 4 (PWPER4)	$006B	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	1111 1111
Pulse width duty 1 (PWDTY1)	$006C	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	1111 1111
Pulse width duty 2 (PWDTY2)	$006D	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	1111 1111
Pulse width duty 3 (PWDTY3)	$006E	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	1111 1111
Pulse width duty 4 (PWDTY4)	$006F	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	1111 1111
SCI baud rate high (SCBDH)	$0070	BTST	BSPL	SYNC	SBR12	SBR11	SBR10	SBR9	SBR8	0000 0000
SCI baud rate low (SCBDL)	$0071	SBR7	SBR6	SBR5	SBR4	SBR3	SBR2	SBR1	SBR0	0000 0100
SCI control 1 (SCCR1)	$0072	LOOPS	WOMS	0	M	WAKE	ILT	PE	PT	0000 0000
SCI control 2 (SCCR2)	$0073	TIE	TCIE	RIE	ILIE	TE	RE	RWU	SBK	0000 0000
SCI status 1 (SCSR1)	$0074	TDRE	TC	RDRF	IDLE	OR	NF	FE	PF	1100 0000
SCI status 2 (SCSR2)	$0075	0	0	0	0	0	0	0	RAF	0000 0000
SCI data high (SCDRH)	$0076	R8	T8	0	0	0	0	0	0	undefined
SCI data low (SCDRL)	$0077	R7T7	R6T6	R5T5	R4T4	R3T3	R2T2	R1T1	R0T0	undefined
Reserved	$0078									
Reserved	$0079									
Reserved	$007A									
Reserved	$007B									
Port H data (PORTH)	$007C	PH7	PH6	PH5	PH4	PH3	PH2	PH1	PH0	undefined
Data direction H (DDRH)	$007D	DDH7	DDH6	DDH5	DDH4	DDH3	DDH2	DDH1	DDH0	0000 0000
Port G data (PORTG)	$007E	PG7	PG6	PG5	PG4	PG3	PG2	PG1	PG0	undefined
Data direction G (DDRG)	$007F	0	0	DDG5	DDG4	DDG3	DDG2	DDG1	DDG0	0000 0000
Timer control register 3 (TCTL3)	$0080	OM1	OL1	OM2	OL2	OM3	OL3	OM4	OL4	0000 0000
Timer control register 4 (TCTL4)	$0081	EDGB	EDGA	PR2B	PR2A	ECEB	ECEA	T2STP	I1/04	0000 0000
Timer 2 counter register (TCNT2) high	$0082	(bit 15)	(14)	(13)	(12)	(11)	(10)	(9)	(bit 8)	0000 0000
Timer 2 counter register (TCNT2) low	$0083	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	0000 0000
Timer 2 output compare 1(T2OC1) high	$0084	(bit 15)	(14)	(13)	(12)	(11)	(10)	(9)	(bit 8)	1111 1111
Timer 2 output compare 1 (T2OC1) low	$0085	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	1111 1111
Timer 2 output comp. 2 (T2OC2) high	$0086	(bit 15)	(14)	(13)	(12)	(11)	(10)	(9)	(bit 8)	1111 1111

Table 4-2 Register and control bit assignments (Page 5 of 5)

Register name	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Timer 2 output comp. 2 (T2OC2) low	$0087	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	1111 1111
Timer 2 output comp. 3 (T20C3) high	$0088	(bit 15)	(14)	(13)	(12)	(11)	(10)	(9)	(bit 8)	1111 1111
Timer 2 output comp. 3(T20C3) low	$0089	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	1111 1111
Timer 2 channel 4 (T2C4) high	$008A	(bit 15)	(14)	(13)	(12)	(11)	(10)	(9)	(bit 8)	1111 1111
Timer 2 channel 4 (T2C4) low	$008B	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	1111 1111
Timer 2 mask (T2MSK)	$008C	OC1I	OC2I	OC3I	C4I	TO2I	0	0	0	0000 0000
Timer 2 flag (T2FLG)	$008D	OC1F	OC2F	OC3F	C4F	TO2F	0	0	0	0000 0000
Port J data (PORTJ)	$008E	PJ7	PJ6	PJ5	PJ4	PJ3	PJ2	PJ1	PJ0	undefined
Data direction J (DDRJ)	$008F	DDJ7	DDJ6	DDJ5	DDJ4	DDJ3	DDJ2	DDJ1	DDJ0	0000 0000
Timer control register 5 (TCTL5)	$0090	OM1	OL1	OM2	OL2	OM3	OL3	OM4	OL4	0000 0000
Timer control register 6 (TCTL6)	$0091	EDGB	EDGA	PR3B	PR3A	ECEB	ECEA	T3STP	I1/O4	0000 0000
Timer 3 counter (TCNT3) high	$0092	(bit 15)	(14)	(13)	(12)	(11)	(10)	(9)	(bit 8)	0000 0000
Timer 3 counter (TCNT3) low	$0093	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	0000 0000
Timer 3 output compare 1 (T30C1) high	$0094	(bit 15)	(14)	(13)	(12)	(11)	(10)	(9)	(bit 8)	1111 1111
Timer 3 output compare 1 (T30C1) low	$0095	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	1111 1111
Timer 3 output compare 2 (T30C2) high	$0096	(bit 15)	(14)	(13)	(12)	(11)	(10)	(9)	(bit 8)	1111 1111
Timer 3 output compare 2 (T30C2) low	$0097	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	1111 1111
Timer 3 output comp. 3 (T3OC3) high	$0098	(bit 15)	(14)	(13)	(12)	(11)	(10)	(9)	(bit 8)	1111 1111
Timer 3 output comp. 3 (T3OC3) low	$0099	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	1111 1111
Timer 3 channel 4 (T3C4) high	$009A	(bit 15)	(14)	(13)	(12)	(11)	(10)	(9)	(bit 8)	1111 1111
Timer 3 channel 4 (T3C4) low	$009B	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	1111 1111
Timer 3 mask (T3MSK)	$009C	OC1I	OC2I	OC3I	C4I	TO3I	0	0	0	0000 0000
Timer 3 flag (T3FLG)	$009D	OC1F	OC2F	OC3F	C4F	TO3F	0	0	0	0000 0000
Port K data (PORTK)	$009E	PK7	PK6	PK5	PK4	PK3	PK2	PK1	PK0	undefined
Data direction K (DDRK)	$009F	DDK7	DDK6	DDK5	DDK4	DDK3	DDK2	DDK1	DDK0	0000 0000

KEY

x State on reset depends on mode selected

u State of bit on reset is undefined

4.3 System initialization

Registers and bits that control initialization and the basic operation of the MCU are protected against writes except under special circumstances. The following table lists registers that can be written only once after reset, or that must be written within the first 64 cycles after reset.

Table 4-3 Registers with limited write access

Register address	Register name	Must be written in first 64 cycles	Write once only
$x024	Timer interrupt mask register 2 (TMSK2)	(1)	—
$x035	Block protect register (BPROT)	(2)	—
$x037	EEPROM mapping register (INIT2)	No	Yes
$x038	System configuration options register 2 (OPT2)	No	(3)
$x039	System configuration options register (OPTION)	(4)	—
$x03D	RAM and I/O map register (INIT)	(5)	—
$x081	Timer control register 4 (Timer 2) TCTL4	No	(6)
$x091	Timer control register 6 (Timer 3) TCTL6	No	(6)

(1) When SMOD = 0, bits 1 and 0 can be written only once, during the first 64 cycles, after which they become read-only. When SMOD = 1, however, these bits can be written at any time. All other bits can be written at any time.

(2) Bits can be written to zero once and only in the first 64 cycles or in special modes. Bits can be set to one at any time.

(3) Bit 4 (IRVNE) can be written only once.

(4) When SMOD = 0, bits 5, 4, 2, 1, and 0 can be written once and only in the first 64 cycles. When SMOD = 1, however, bits 5, 4, 2, 1, and 0 can be written at any time. All other bits can be written at any time.

(5) When SMOD = 0, bits can be written only once, during the first 64 cycles, after which the register becomes read-only. When SMOD = 1, bits can be written at any time.

(6) Bits 5, 4, 3 and 2 can be written only once.

4.3.1 Mode selection

The four mode variations are selected by the logic states of the mode A (MODA) and mode B (MODB) pins during reset. The MODA and MODB logic levels determine the logic state of the special mode (SMOD) and mode A (MDA) control bits in the highest priority I-bit interrupt and miscellaneous (HPRIO) register.

After reset is released, the mode select pins no longer influence the MCU operating mode. In single chip operating mode, MODA pin is connected to a logic zero. In expanded mode, MODA is normally connected to V_{DD} through a pull-up resistor of 4.7 kΩ. The MODA pin also functions as the load instruction register ($\overline{LIR}$) pin when the MCU is not in reset. The open-drain active low $\overline{LIR}$ output pin drives low during the first E cycle of each instruction. The MODB pin also functions as the stand-by power input (VSTBY), which allows the RAM contents to be maintained in the absence of V_{DD}.

Refer to Table 4-4, which is a summary of mode pin operation, the mode control bits and the four operating modes.

A normal mode is selected when MODB is logic one during reset. One of three reset vectors is fetched from address $FFFA–$FFFF, and program execution begins from the address indicated by this vector. If MODB is logic zero during reset, the special mode reset vector is fetched from addresses $BFFA–$BFFF and software has access to special test features. Refer to Section 5.

4.3.1.1 HPRIO — Highest priority I-bit interrupt & misc. register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Highest priority interrupt (HPRIO)	$003C	RBOOT	SMOD	MDA	PSEL4	PSEL3	PSEL2	PSEL1	PSEL0	xxx0 0110

Note: RBOOT, SMOD and MDA bits depend on the power-up initialization mode and can only be written in special modes when SMOD = 1. Refer to Table 4-4.

RBOOT — Read bootstrap ROM

1 (set) – Bootloader ROM enabled, at $BE40–$BFFF.

0 (clear) – Bootloader ROM disabled and not in map.

SMOD — Special mode select

1 (set) – Special mode variation in effect.

0 (clear) – Normal mode variation in effect.

Once cleared, cannot be set again.

MDA — Mode select A

1 (set) – Normal expanded or special test mode. (Expanded buses active.)

0 (clear) – Normal single chip or special bootstrap mode. (Ports active.)

Table 4-4 Hardware mode select summary

Inputs		Mode	Control bits in HPRIO (latched at reset)		
MODB	MODA		RBOOT	SMOD	MDA
1	0	Single chip	0	0	0
1	1	Expanded	0	0	1
0	0	Special bootstrap	1	1	0
0	1	Special test	0	1	1

PSEL[4:0] — Priority select bits (refer to Section 5)

4.3.2 Initialization

Because bits in the following registers control the basic configuration of the MCU, an accidental change of their values could cause serious system problems. The protection mechanism, overridden in special operating modes, requires a write to the protected bits only within the first 64 bus cycles after any reset, or only once after each reset. See Table 4-3.

4

4.3.2.1 CONFIG — System configuration register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Configuration control (CONFIG)	$003F	1	1	CLKX	PAREN	NOSEC	NOCOP	1	EEON	11xx xx1x

CONFIG controls the presence of EEPROM in the memory map and enables the COP watchdog system. The CLKX bit enables the XOUT pin to output the XCLK signal, and the PAREN bit enables pull-ups on certain ports. Refer to Section 4.6.3. A security feature that protects data in EEPROM and RAM is available, controlled by the NOSEC bit (refer to Section 4.6.3).

CONFIG is made up of EEPROM cells and static working latches. The operation of the MCU is controlled directly by these latches and not the EEPROM byte. When programming the CONFIG register, the EEPROM byte is accessed. When the CONFIG register is read, the static latches are accessed.

These bits can be read at any time. The value read is the one latched into the register from the EEPROM cells during the last reset sequence. A new value programmed into this register is not readable until after a subsequent reset sequence.

Bits in CONFIG can be written at any time if SMOD = 1 (bootstrap or special test mode). If SMOD = 0 (single chip or expanded mode), these bits can only be written using the EEPROM programming sequence, and none of the bits are readable or active until latched via the next reset.

Bits [7, 6, 1] — Not implemented; always read as one.

CLKX — XOUT enable

1 (set) – XCLK signal is driven out on the XOUT pin.

0 (clear) – XOUT pin is disabled.

The frequency of the XCLK signal is controlled by two bits in the OPT2 register (see Section 4.3.2.5).

PAREN — Pull-up assignment register enable (refer to Section 6)

1 (set) – Pull-ups can be enabled using PPAR.

0 (clear) – All pull-ups disabled (not controlled by PPAR).

NOSEC — EEPROM security disabled (refer to Section 4.6.3)

1 (set) – Disable security.

0 (clear) – Enable security.

NOCOP — COP system disable (refer to Section 5)

1 (set) – COP system disabled.

0 (clear) – COP system enabled (forces reset on timeout).

EEON — EEPROM enable

1 (set) – EEPROM included in the memory map.

0 (clear) – EEPROM is excluded from the memory map.

4.3.2.2 INIT — RAM and I/O mapping register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
RAM & I/O mapping (INIT)	$003D	RAM3	RAM2	RAM1	RAM0	REG3	REG2	REG1	REG0	0000 0000

The internal registers used to control the operation of the MCU can be relocated on 4K boundaries within the memory space with the use of INIT. This 8-bit special-purpose register can change the default locations of the RAM and control registers within the MCU memory map. It can be written to only once within the first 64 E clock cycles after a reset. It then becomes a read-only register.

RAM[3:0] — RAM map position

These four bits, which specify the upper hexadecimal digit of the RAM address, control the position of the RAM in the memory map. The RAM can be positioned at the beginning of any 4K page in the memory map. Refer to Table 4-5.

REG[3:0] — 160-byte register block position

These four bits specify the upper hexadecimal digit of the address for the 160-byte block of internal registers. The register block is positioned at the beginning of any 4K page in the memory map. Refer to Table 4-5.

Table 4-5 RAM and register remapping

RAM[3:0]	Location
0000	$0000–$02FF
0001	$1000–$12FF
0010	$2000–$22FF
0011	$3000–$32FF
0100	$4000–$42FF
0101	$5000–$52FF
0110	$6000–$62FF
0111	$7000–$72FF
1000	$8000–$82FF
1001	$9000–$92FF
1010	$A000–$A2FF
1011	$B000–$B2FF
1100	$C000–$C2FF
1101	$D000–$D2FF
1110	$E000–$E2FF
1111	$F000–$F2FF

REG[3:0]	Location
0000	$0000–$009F
0001	$1000–$109F
0010	$2000–$209F
0011	$3000–$309F
0100	$4000–$409F
0101	$5000–$509F
0110	$6000–$609F
0111	$7000–$709F
1000	$8000–$809F
1001	$9000–$909F
1010	$A000–$A09F
1011	$B000–$B09F
1100	$C000–$C09F
1101	$D000–$D09F
1110	$E000–$E09F
1111	$F000–$F09F

When the memory map has the 160-byte register block mapped at the same location as RAM, the registers have priority and the RAM is relocated to the memory space immediately following the register block. This mapping feature keeps all the RAM available for use. Refer to Figure 4-2, which illustrates the overlap.

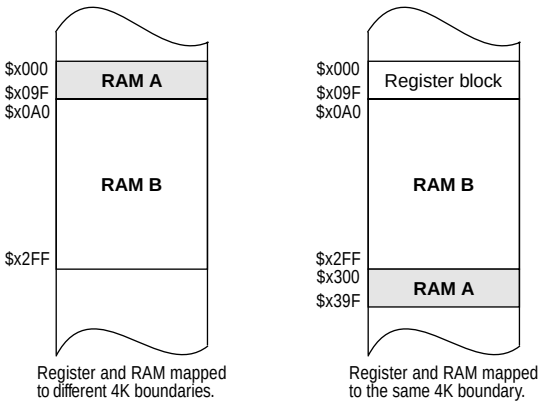


Figure 4-2 RAM and register overlap

4.3.2.3 INIT2 — EEPROM mapping register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
EEPROM mapping (INIT2)	$0037	EE3	EE2	EE1	EE0	0	0	0	0	0000 0000

This register determines the location of EEPROM in the memory map. INIT2 may be read at any time but bits 7–4 may be written only once after reset in normal modes.

EE[3:0] — EEPROM map position

EEPROM is located at $xD80–$xFFF, where x is the hexadecimal digit represented by EE[3:0]. Refer to Table 4-6.

Table 4-6 EEPROM remapping

EE[3:0]	Location	EE[3:0]	Location	EE[3:0]	Location	EE[3:0]	Location
0000	$0D80–$0FFF	0100	$4D80–$4FFF	1000	$8D80–$8FFF	1100	$CD80–$CFFF
0001	$1D80–$1FFF	0101	$5D80–$5FFF	1001	$9D80–$9FFF	1101	$DD80–$DFFF
0010	$2D80–$2FFF	0110	$6D80–$6FFF	1010	$AD80–$AFFF	1110	$ED80–$EFFF
0011	$3D80–$3FFF	0111	$7D80–$7FFF	1011	$BD80–$BFFF	1111	$FD80–$FFFF

Bits [3:0] — Not implemented; always read zero.

4.3.2.4 OPTION — System configuration options register 1

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
System config. options 1 (OPTION)	$0039	ADPU	CSEL	IRQE	DLY	CME	FCME	CR1	CR0	0001 0000

The 8-bit special-purpose OPTION register sets internal system configuration options during initialization. The time protected control bits IRQE, DLY, FCME and CR[1:0] can be written only once in the first 64 cycles after a reset and then they become read-only bits. This minimizes the possibility of any accidental changes to the system configuration. They may be written at any time in special modes.

ADPU — A/D power-up (refer to Section 10)

1 (set) – A/D system power enabled.

0 (clear) – A/D system disabled, to reduce supply current.

After enabling the A/D power, at least 100µs should be allowed for system stabilization.

CSEL — Clock select (refer to Section 10)

1 (set) – A/D and EEPROM use internal RC clock source (about 1.5MHz).

0 (clear) – A/D and EEPROM use system E clock (must be at least 1MHz).

This bit selects the clock source for the on-chip EEPROM and A/D charge pumps. The on-chip RC clock should be used when the E clock frequency falls below 1MHz.

IRQE — Configure $\overline{\text{IRQ}}$ for falling-edge-sensitive operation

1 (set) – Falling-edge-sensitive operation.

0 (clear) – Low-level-sensitive operation.

DLY — Enable oscillator start-up delay

1 (set) – A stabilization delay of around 4064 bus cycles is imposed as the MCU is started up from STOP mode (or power-on reset).

0 (clear) – The oscillator start-up delay is bypassed and the MCU resumes processing within about four bus cycles. A stable external oscillator is required if this option is selected.

DLY is set on reset, so a delay is always imposed as the MCU is started up from power-on reset.

CME — Clock monitor enable (refer to Section 5)

1 (set) – Clock monitor enabled.

0 (clear) – Clock monitor disabled.

In order to use both STOP and clock monitor, the CME bit should be cleared before executing STOP, then set after recovering from STOP.

FCME — Force clock monitor enable (refer to Section 5)

1 (set) – Clock monitor enabled; cannot be disabled until next reset.

0 (clear) – Clock monitor follows the state of the CME bit.

When FCME is set, slow or stopped clocks will cause a clock failure reset sequence. To utilize STOP mode, FCME should always be cleared.

CR[1:0] — COP timer rate select bits (refer to Section 5)

These control bits determine a scaling factor for the watchdog timer.

4.3.2.5 OPT2 — System configuration options register 2

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
System config. options 2 (OPT2)	$0038	LIRDV	CWOM	0	IRVNE	LSBF	SPR2	XDV1	XDV2	000x 0000

LIRDV — $\overline{\text{LIR}}$ driven

1 (set) – Enable $\overline{\text{LIR}}$ drive high pulse.

0 (clear) – $\overline{\text{LIR}}$ not driven high on MODA/$\overline{\text{LIR}}$ pin.

In single-chip and bootstrap modes, this bit has no meaning or effect. The $\overline{\text{LIR}}$ pin is driven low to indicate that execution of an instruction has begun. The $\overline{\text{LIR}}$ pin is normally configured for wired-OR operation (only pulls low). In order to detect consecutive instructions in a high-speed application, this signal can be made to drive high for a quarter of a cycle to prevent false triggering (LIRDV set).

CWOM — Port C wired-OR mode (refer to Section 6)

1 (set) – Port C outputs are open-drain.

0 (clear) – Port C operates normally.

Bits [5, 0] — Not implemented; always read zero.

IRVNE — Internal read visibility/not E

IRVNE may be written once in normal modes, and can be written as often as desired in bootstrap and special test modes. In special test modes, IRVNE is reset to one. In normal and bootstrap modes, IRVNE is reset to zero. IRVNE should only be used at room temperature and 5V nominal.

In **expanded modes**, IRVNE determines whether internal read visibility (IRV) is on or off.

1 (set) – Data from internal reads is driven out of the external data bus.

0 (clear) – No visibility of internal reads on external bus.

In **single chip modes** this bit determines whether the E clock drives out from the chip.

1 (set) – E pin is driven low.

0 (clear) – E clock is driven out from the chip.

Refer to the following table for a summary of the operation immediately following reset.

Mode	IRVNE after reset	E clock after reset	IRV after reset	IRVNE affects only	IRVNE can be written
Single chip	0	On	Off	E	Once
Expanded	0	On	Off	IRV	Once
Boot	0	On	Off	E	Unlimited
Special test	1	On	On	IRV	Unlimited

LSBF — LSB-first enable (refer to Section 8)

1 (set) – Data is transferred LSB first.

0 (clear) – Data is transferred MSB first.

SPR2 — SPI clock rate select (refer to Section 8)

This bit adds a divide-by-four to the SPI clock chain.

XDV[1, 0] — XOUT clock divide select

These two bits control the frequency of the XCLK signal, which is output on the XOUT pin if enabled by the CLKX bit in CONFIG. Table 4-7 shows some example frequencies. Once a clock rate has been selected, a maximum time of 16 E clock cycles should be allowed for the signal to stabilize. Note that on reset, both bits are cleared and the XCLK signal runs at the same frequency as EXTAL.

Note: The phase relationship between XOUT and EXTAL or E cannot be predicted.

Table 4-7 XCLK frequencies

XDV 1	XDV2	EXTAL divided by	XCLK with EXTAL = 16 MHz
0	0	1	16 MHz
0	1	4	4 MHz
1	0	6	2.7 MHz
1	1	8	2 MHz

4.3.2.6 BPROT — Block protect register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Block protect (BPROT)	$0035	BULKP	BIT6	BPRT4	PTCON	BPRT3	BPRT2	BPRT1	BPRT0	1111 1111

BPROT prevents accidental writes to EEPROM and the CONFIG register, and enables the low voltage EEPROM protect circuit. The bits in this register can be written to zero only once during

the first 64 E clock cycles after reset in the normal modes; they can be set at any time. Once the bits are cleared, the EEPROM array and the CONFIG register can be programmed or erased. Setting the bits in the BPROT register to logic one protects the EEPROM and CONFIG register until the next reset. Refer to Table 4-8.

BULKP — Bulk erase of EEPROM protect

1 (set) – EEPROM cannot be bulk or row erased.

0 (clear) – EEPROM can be bulk erased normally.

BIT6

BIT6 can be programmed to 0 in the first 64 cycles, although the bit has no meaning.

PTCON — Protect for CONFIG register

1 (set) – CONFIG register cannot be programmed or erased.

0 (clear) – CONFIG register can be programmed or erased normally.

Note that, in special modes, CONFIG may be written regardless of the state of PTCON.

BPRT[4:0] — Block protect bits for EEPROM

1 (set) – Protection is enabled for associated block; it cannot be programmed or erased.

0 (clear) – Protection disabled for associated block.

Each of these five bits protects a block of EEPROM against writing or erasure, as follows:

Table 4-8 EEPROM block protect

Bit name	Block protected	Block size
BPRT0	$xD80–$xD9F	32 bytes
BPRT1	$xDA0–$xDDF	64 bytes
BPRT2	$xDE0–$xE5F	128 bytes
BPRT3	$xE60–$xF7F	288 bytes
BPRT4	$xF80–$xFFF	128 bytes

4.3.2.7 TMSK2 — Timer interrupt mask register 2

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Timer interrupt mask 2 (TMSK2)	$0024	TOI	RTII	PAOVI	PAII	0	0	PR1	PR0	0000 0000

PR[1:0] are time-protected control bits and can be changed only once and then only within the first 64 bus cycles after reset in normal modes.

Note: Bits [7:4] in TMSK2 correspond bit for bit with the flag bits in TFLG2. Ones in bits [7:4] of TMSK2 enable the corresponding interrupt sources.

TOI — Timer overflow interrupt enable (refer to Section 9)

1 (set) – Interrupt requested when TOF is set.

0 (clear) – TOF interrupts disabled.

RTII — Real-time interrupt enable (refer to Section 9)

1 (set) – Interrupt requested when RTIF set.

0 (clear) – RTIF interrupts disabled.

PAOVI — Pulse accumulator overflow interrupt enable (refer to Section 9)

1 (set) – Intdrrupt requested when PAOVF set.

0 (clear) – PAOVF interrupts disabled.

PAII — Pulse accumulator interrupt enable (refer to Section 9)

1 (set) – Interrupt requested when PAIF set.

0 (clear) – PAIF interrupts disabled.

Bits [3, 2] — Not implemented; always read zero.

PR[1:0] — Timer prescaler select

These two bits select the prescale rate for the main 16-bit free-running timer system, Timer 1. These bits can be written only once during the first 64 E clock cycles after reset in normal modes, or at any time in special modes. Refer to the following table:

PR[1:0]	Prescale factor
0 0	1
0 1	4
1 0	8
1 1	16

4.3.2.8 TCTL4 and TCTL6 — Timer 2 and 3 control registers

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Timer control register 4 (TCTL4)	$0081	EDGB	EDGA	PR2B	PR2A	ECEB	ECEA	T2STP	I1/04	0000 0000

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Timer control register 6 (TCTL6)	$0091	EDGB	EDGA	PR3B	PR3A	ECEB	ECEA	T3STP	I1/04	0000 0000

Bits [5:2] in both these registers can be written only once after reset. The following paragraphs describe the Timer 2 control bits in TCTL4; the Timer 3 control bits in TCTL6 are described in Section 9.

EDGB and EDGA — Input capture edge control (Refer to Section 9)

This pair of bits configures the input capture edge detector circuits for IC1. IC1 functions only if the I1/O4 bit is set.

PR2A and PR2B — Timer 2 prescaler select

These bits are used to select the prescaler divide-by ratio for Timer 2. They can be written to only once after reset.

PR2B	PR2A	Prescaler
0	0	1
0	1	4
1	0	8
1	1	16

PR3A and PR3B — Timer 3 prescaler select

These bits are used to select the prescaler divide-by ratio for Timer 3. They can only be written to once after reset. If PR3B and PR3A are both cleared, then Timer 3 is synchronized to the prescaled Timer 1 rate.

PR3B	PR3A	Prescaler
0	0	Use Timer 1 rate
0	1	1
1	0	4
1	1	16

ECEB and ECEA — Event counter edge control

These control bits configure the input clock source for the Timer 2 counter. They can be written to only once after reset.

ECEB	ECEA	Configuration
0	0	Timer 2 uses internal clock and prescaler
0	1	Count on rising edges of external clock only
1	0	Count on falling edges of external clock only
1	1	Count on any edge of external clock

T2STP — Stop Timer 2 counter (Refer to Section 9)

1 (set) – Timer 2 counter and prescaler are stopped and the counter is reset to $0000.

0 (clear) – Timer 2 counter operates normally.

I1/O4 — Input capture 1/output compare 4 (Refer to Section 9)

1 (set) – Input capture 1 function is enabled (no OC4).

0 (clear) – Output compare 4 function is enabled (no IC1).

4.4 Memory expansion

4.4.1 Memory expansion logic

The MC68HC11KW1 has the ability to extend the address range of the M68HC11 CPU beyond the physical 64K byte limit of the 16 CPU address lines. The extra addressing capability is provided by a register-based paging scheme using expansion address lines and the physical 64K bytes of CPU address space.

Two additional on-chip blocks are provided with the MC68HC11KW1. The first block implements additional address lines that become active only when required by the CPU. The second block provides chip-select signals that simplify the interface to external peripheral devices. Both of these blocks are fully programmable by values written to associated control registers.

4.4.2 Extended addressing

Memory expansion is achieved by manipulating the CPU address lines such that, even though the CPU cannot distinguish more than 64K bytes of physical memory, up to 1M byte can be accessed through a paged memory scheme. Additional address lines XA[18:13] are provided as alternative functions of port G pins. Bits in the port G assignment register (PGAR) define which port G pins are to be used for memory expansion address lines and which are to be used for general-purpose I/O.

In order to access expanded memory, the user must first allocate a range of the 64K byte address space to be used for the window(s) through which external, expanded memory is viewed by the CPU. The size and placement of the window(s) depend on values written to the MMSIZ and MMWBR registers, respectively. Which bank or page of the expanded memory that is present in the window(s) at a given time is dependent on values written to the MM1CR and MM2CR registers.

Up to two windows can be designated and each can be programmed to 0K (disabled), 8K, 16K, or 32K bytes. The base address for each window must be an integer multiple of the window size, with the exception of the 32K byte window, for which the base address can be at $0000, $4000, or $8000.

If the windows are defined in such a way that they overlap, bank window 1 has priority and the part of window 2 that is not overlapped by bank window 1 remains active. If a window is defined such that it overlaps any internal registers, RAM, or EEPROM, then the portion of the registers, RAM, or EEPROM that is overlapped is repeated in all banks associated with that window.

Coming out of reset, the reset vector is fetched from external memory. Since the memory expansion lines are disabled coming out of reset and can be internally pulled to logic level one, any external system that uses these expansion address lines sees them as all ones. In this case, the reset vector is fetched from $7FFFE–$7FFFF. Systems using external but not expanded memory still fetch the reset vector from $FFFE–$FFFF. This is the reset vector's normal position at the top of the M68HC11 CPU's conventional 64K byte address space.

Expanded memory is addressed by using a combination of the CPU's normal address lines ADDR[15:0] and the expansion address lines XA[18:13]. Window size and the number of banks associated with the window determine exactly which address lines are used. The additional address lines (XA[18:13]) determine which bank is present in a window at a given time. The lower three expansion address lines (XA[15:13]) are used only when needed by the CPU and replace the CPU's equivalent address lines (ADDR[15:13]). Table 4-9 shows which address lines are used for various configurations of expanded memory.

A special case exists when the bank size is 32K bytes and the window base address is $4000. Normally, when the bank size is 32K bytes and the bank address is $0000 or $8000, CPU address lines ADDR[14:0] select individual bytes within the 32K byte space and the ADDR[14:0] pins are connected to address lines (A[14:0]) of the memory device. When the base address is $4000, the CPU address signal ADDR14 must be inverted to allow 32K bytes of contiguous memory. The MC68HC11KW1 CPU drives the inverted CPU ADDR14 signal onto the XA14 pin when the window is active. In this case, the XA14 signal must be connected to the address line 14 of the memory device. When the window is not active, the XA14 pin is driven with the non-inverted CPU ADDR14 signal.

Table 4-9 CPU address and address expansion signals

Number of banks	Window size			
	8K bytes	16K bytes	32K bytes	32K bytes (window based at $4000)
2	ADDR[12:0]	ADDR[13:0]	ADDR[14:0]	ADDR[13:0]
	XA13	XA14	XA15	XA[15:14]
4	ADDR[12:0]	ADDR[13:0]	ADDR[14:0]	ADDR[13:0]
	XA[14:13]	XA[15:14]	XA[16:15]	XA[16:14]
8	ADDR[12:0]	ADDR[13:0]	ADDR[14:0]	ADDR[13:0]
	XA[15:13]	XA[16:14]	XA[17:15]	XA[17:14]
16	ADDR[12:0]	ADDR[13:0]	ADDR[14:0]	ADDR[13:0]
	XA[16:13]	XA[17:14]	XA[18:15]	XA[18:14]
32	ADDR[12:0]	ADDR[13:0]	—	—
	XA[17:13]	XA[18:14]	—	—
64	ADDR[12:0]	—	—	—
	XA[18:13]	—	—	—

If neither bank uses a particular expansion address bit, the corresponding pin is available for general-purpose I/O. The PGAR register selects which pins are used for I/O or memory expansion address lines.

4.4.3 Memory expansion examples

Consider an example system in which an external memory is used and the user wishes to allocate a single 8K byte window through which to access a total of 64K bytes of external memory. To provide the 8K byte address range needed for the window, only CPU address lines ADDR[12:0] need be used to provide 8K bytes (2^{13}) address locations. Expansion address lines XA[15:13] replace CPU address lines ADDR[15:13] and provide an additional eight times (2^3) the number of address locations provided by ADDR[12:0], (a total of 64K bytes of address space). ADDR[12:0] provide the 8K byte window and XA[15:13] provide an additional eight bank-select signals that determine which bank is present in the window. This is illustrated inFigure 4-3 and Figure 4-4. Figure 4-3 shows a memory map and Figure 4-4 shows a schematic for a single 8K byte window with 8 banks of external memory.

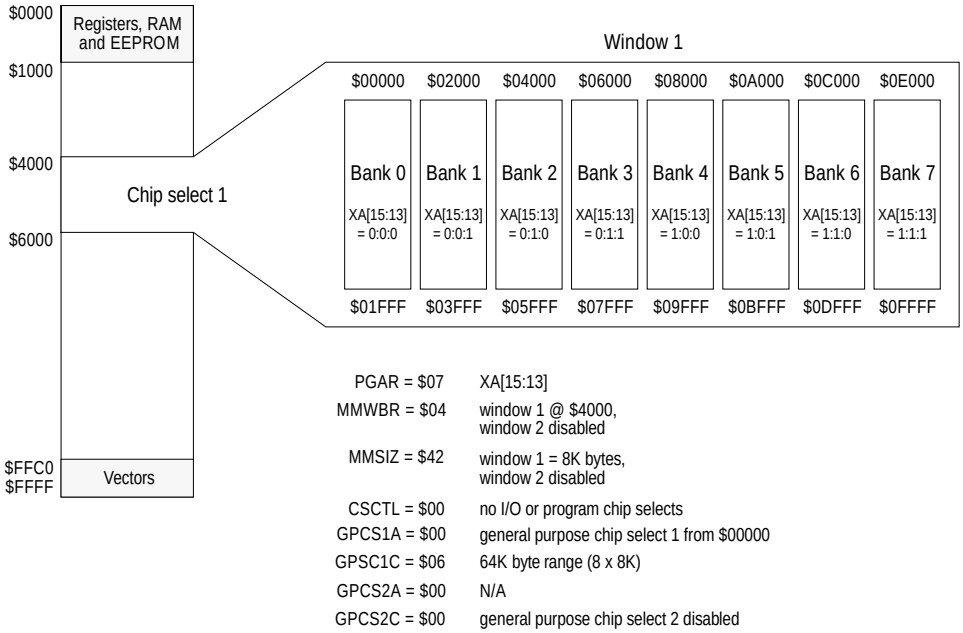


Figure 4-3 Memory map example of memory expansion

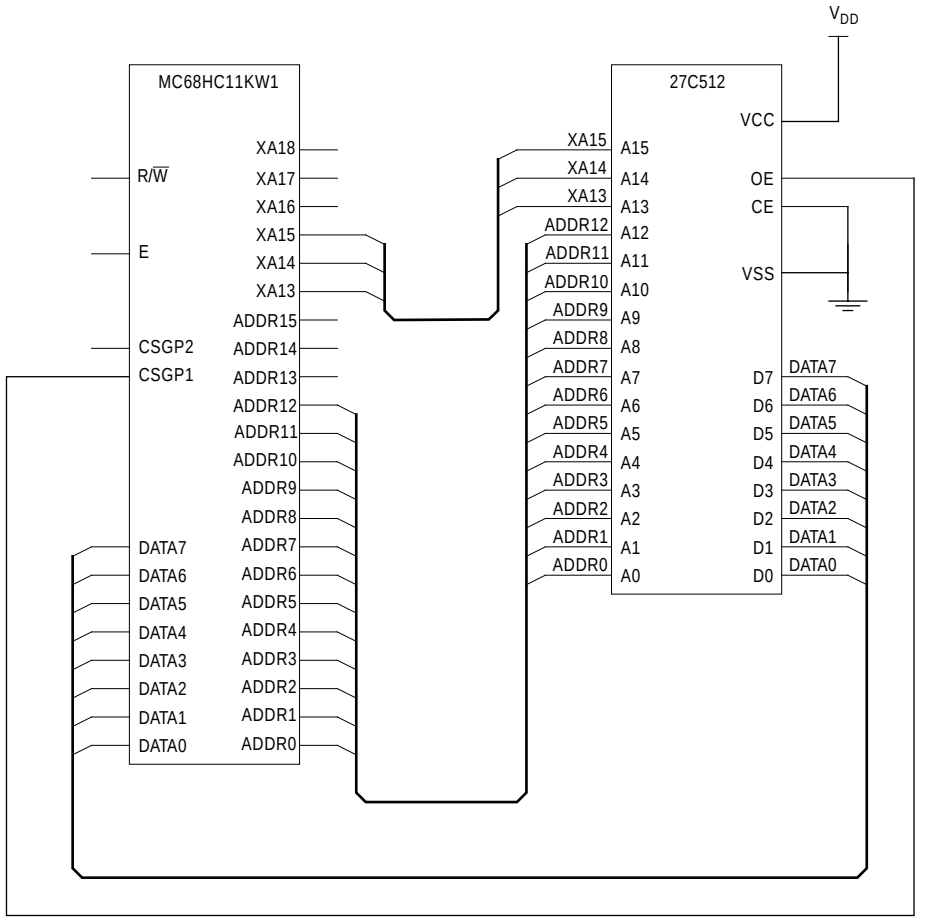


Figure 4-4 Schematic example of memory expansion

In another example the user wishes to allocate two windows. The first window is organized as in the previous example, 8 banks of 8K bytes each. The second window is organized as 16 banks of 16K bytes each. The logical addresses in window 2 are determined by CPU address lines ADDR[13:0]. Note that XA13 replaces ADDR13 for each memory device in this example since ADDR13 is driven on XA13. Expansion address lines XA[17:14] add another 16 (2^4) times the number of address locations provided by ADDR[13:0] (256K bytes total address space). ADDR13 may also be used instead of XA13 for the 6226 memory devices if the designer chooses. This is illustrated in Figure 4-3 and Figure 4-6. Figure 4-3 shows a memory map and Figure 4-6 shows a schematic for one 8K byte window with 8 banks of external memory, and one 16K byte window with 16 banks of external memory.

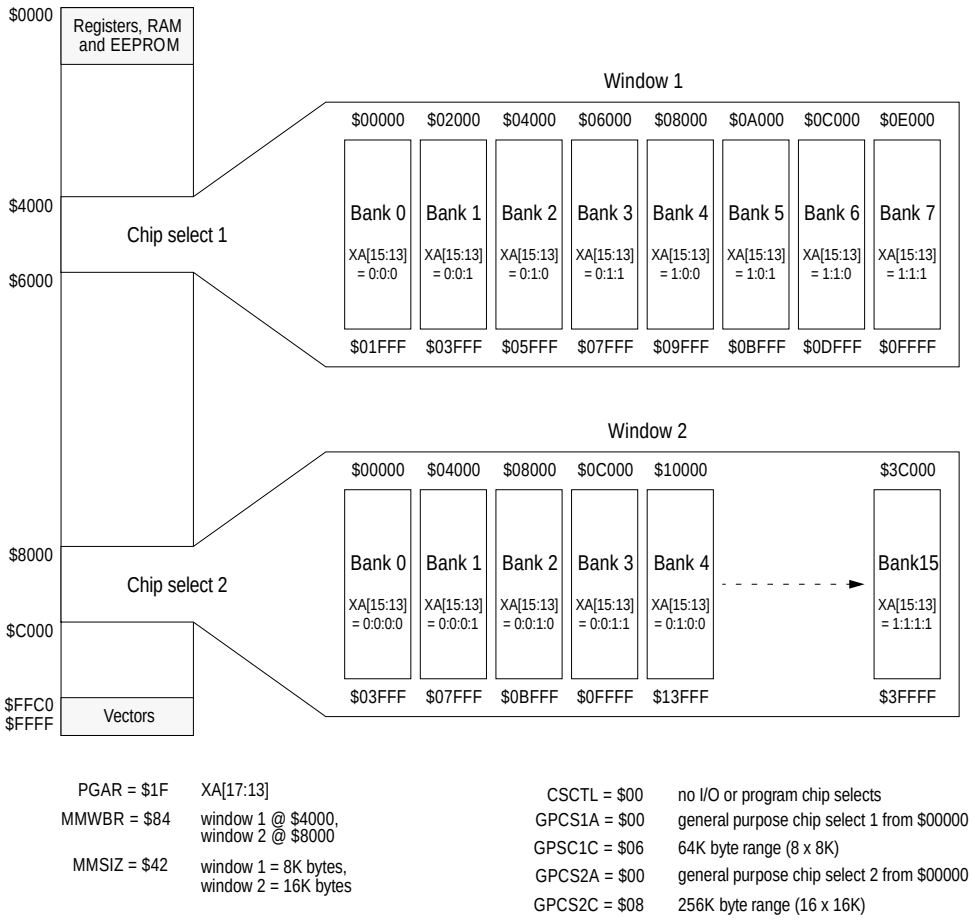


Figure 4-5 Memory map example of memory expansion

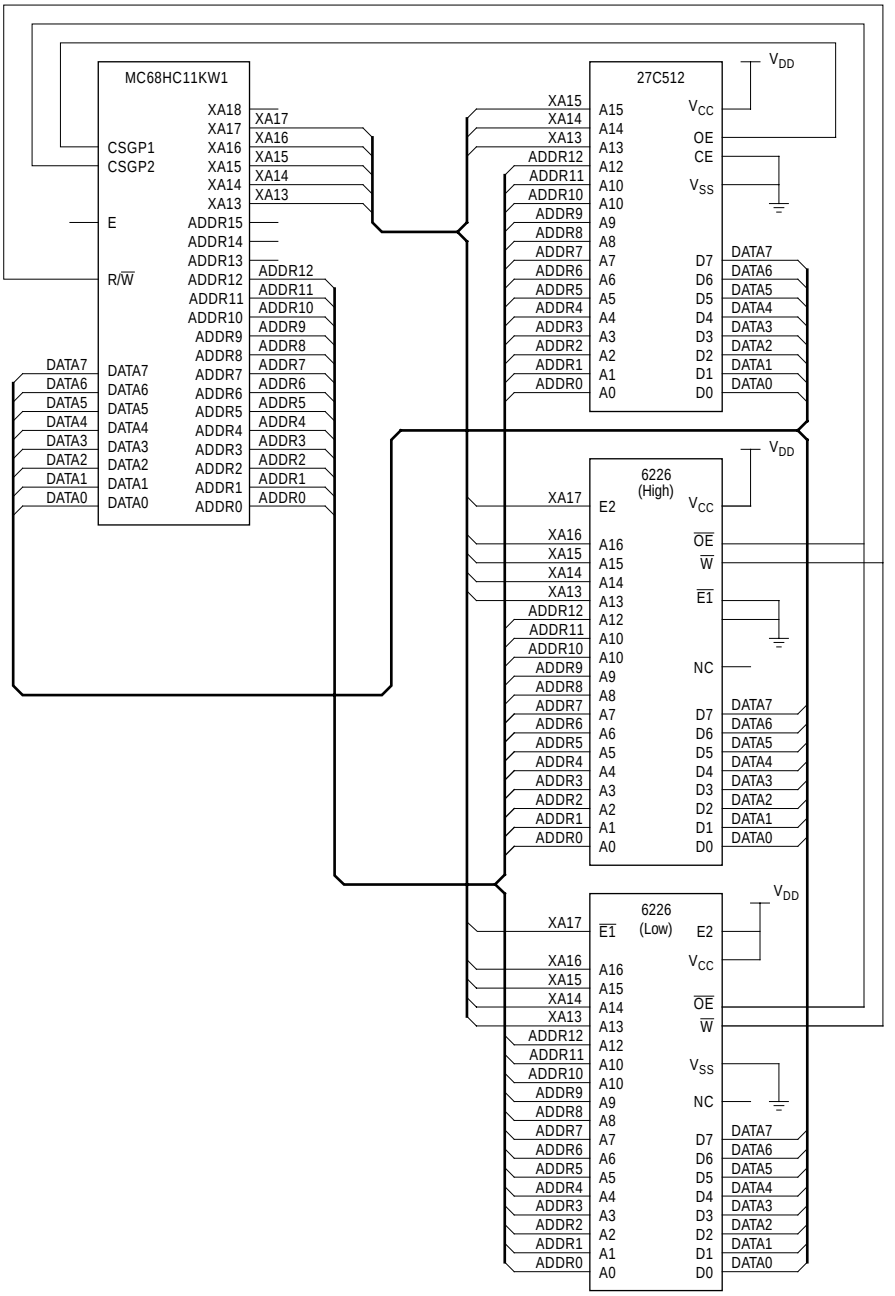


Figure 4-6 Schematic example of memory expansion

4.4.4 MMSIZ — Memory mapping window size register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Memory mapping window size (MMSIZ)	$0056	MXGS2	MXGS1	W2SZ1	W2SZ0	0	0	W1SZ1	W1SZ0	0000 0000

The MMSIZ register sets the size of the windows in use and selects whether the on-board general-purpose chip selects are active for CPU addresses or for expansion addresses.

MXGS[2:1] — Memory expansion select for general-purpose chip select 2 or 1

1 (set) – General-purpose chip select 2 or 1 based on expansion address.

0 (clear) – General-purpose chip select 2 or 1 based on 64K byte CPU address.

W2SZ[1:0] — Window 2 size

These bits select the bank size for window 2. The window starting address depends on the contents of the MMWBR register and continues for the same number of bytes as the selected window size. Refer to Table 4-10.

Bits 3 and 2 — Not implemented; always read zero.

W1SZ[1:0] — Window 1 size

These bits select the bank size for window 1. The window starting address depends on the contents of the MMWBR register and continues for the same number of bytes as the selected window size. Refer to Table 4-10.

Table 4-10 Window size select

WxSZ[1:0]	Window size
0 0	Window disabled
0 1	8K – window can have up to 64 8K byte banks
1 0	16K – window can have up to 32 16K byte banks
1 1	32K – window can have up to 16 32K byte banks

4.4.5 MMWBR – Memory mapping window base register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Memory mapping window base (MMSIZ)	$0057	W2A15	W2A14	W2A13	0	W1A15	W1A14	W1A13	0	0000 0000

The MMWBR register defines the starting address of each of the two windows within the CPU 64K byte address range. The windows normally begin at a boundary related to their size (an 8K byte window can begin on any 8K byte boundary, beginning at $0000).

W2A[15:13] — Window 2 base address

These bits select the three most significant bits (MSB) of the base address for memory mapping window 2. Note that W2A13 is ignored if the bank size is set for 16 or 32K bytes. Refer to Figure 4-11.

Bits 4 and 0 — Not implemented; always read zero.

W1A[15:13] — Window base 1 address

These bits select the three MSBs of the base address for memory mapping window 1. Note that W1A13 is ignored if the bank size is set for 16 or 32K bytes. Refer to Table 4-11.

Table 4-11 Memory expansion window base address

MSB bits	Window base address		
WxA[15:13]	8K bytes	16K bytes	32K bytes
0 0 0	$0000	$0000	$0000
0 0 1	$2000	$0000	$0000
0 1 0	$4000	$4000	$4000
0 1 1	$6000	$4000	$4000
1 0 0	$8000	$8000	$8000
1 0 1	$A000	$8000	$8000
1 1 0	$C000	$C000	$8000
1 1 1	$E000	$C000	$8000

4.4.6 MM1CR, MM2CR – Memory mapping window 1 and 2 control registers

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Memory mapping window 1 control (MM1CR)	$0058	0	X1A18	X1A17	X1A16	X1A15	X1A14	X1A13	0	0000 0000

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Memory mapping window 2 control (MM2CR)	$0059	0	X2A18	X2A17	X2A16	X2A15	X2A14	X2A13	0	0000 0000

These two window registers indicate which bank of a window is active. Each contains the value to be output when the CPU selects addresses within the extended memory window. To change banks, write the address of the new bank into the appropriate window register.

Bits 7 and 0 — Not implemented; always read zero.

MM1CR — Memory mapping window 1 control register

When a 64K byte CPU address falls within window 1, the value in MM1CR is driven out from the corresponding expansion address lines to enable the specified bank in the window.

MM2CR — Memory mapping window 2 control register

When a 64K byte CPU address falls within window 2, the value in MM2CR is driven out from the corresponding expansion address lines to enable the specified bank in the window.

Overlap guidelines:

- On-chip registers, RAM, and EEPROM have higher priority than expansion windows. If a window overlaps RAM, registers or EEPROM, they appear in all banks at their CPU address.
- Window 1 has a higher priority than window 2, therefore any overlapped portion of window 2 is inaccessible.

4.4.7 PGAR — Port G assignment register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Port G assignment (PGAR)	$002D	0	0	PGAR5	PAGR4	PGAR3	PGAR2	PGAR1	PGAR0	0000 0000

PGAR selects which pins are used for I/O or memory expansion address lines, defining which extended address lines are used. The memory expansion address lines are shared with port G I/O pins. Selecting an address on one of these pins causes a port G pin to be lost. Therefore, to allow unused lines to serve as general-purpose I/O, select only those address lines that are needed by the expansion logic. If neither bank uses a particular expansion address bit, the corresponding pin is available for general-purpose I/O. If an address line is not required, clear the appropriate bit in PGAR. (A special case exists for the address lines that overlap the CPU address lines XA[15:13]. If these lines are selected as address lines in PGAR, but are not used in either window, the corresponding CPU address line is output on the appropriate port G pin.)

Bits [7:6] — Not implemented; always read zero.

PGAR[5:0] — Port G pin assignment

1 (set) – Corresponding port G pin is expansion address line (XA[18:13]).

0 (clear) – Corresponding port G pin is general-purpose I/O.

4.5 Chip selects

The function of the chip selects is to minimize the amount of external glue logic needed to interface the MCU to external devices. Such factors as polarity, address block size, and clock stretching can be controlled using the chip-select registers.

When enabled, a chip select signal is asserted whenever the CPU makes an access to a designated range of addresses. Bus control signals and chip select signals are synchronous with the external E clock signal. Refer to the section on expansion bus timing (Section A.5.4) in the electrical specifications chapter. The length of the external E clock cycle to which the external device is synchronized can be stretched to accommodate devices that are slower than the MCU.

There are six chip select control registers. Chip select functions are enabled by control bits in the CSCTL register. When an MCU pin is not used for chip select functions, it can be used for general-purpose I/O.

The MC68HC11KW1 has four software configured chip selects that are enabled in expanded modes. The chip select for I/O (CSIO) is used for I/O expansion. The program chip select ($\overline{\text{CSPROG}}$) is used with an external memory that contains the reset vectors and program. The two general-purpose chip selects, CSGP1 and CSGP2, are used to enable external devices. These external devices can be in the 64K byte memory space or in the expanded memory space.

4.5.1 Chip select priorities

To minimize conflict between chip selects with one another or with internal memory and registers, priority is determined by the GPSPR bit in the CSCTL register. Refer to Figure 4-12.

Table 4-12 Chip select priorities

GCSPR = 0	GCSPR = 1
On-chip registers	On-chip registers
On-chip RAM	On-chip RAM
Bootloader ROM	Bootloader ROM
On-chip EEPROM	On-chip EEPROM
I/O chip select	I/O chip select
Program chip select	GP chip select 1
GP chip select 1	GP chip select 2
GP chip select 2	Program chip select

4.5.2 Program chip select

The program chip select ($\overline{\text{CSPROG}}$) is active in the range of memory where the main program exists. Other chip selects are active when their respective memory areas are used. Refer to Table 4-13.

$\overline{\text{CSPROG}}$ is enabled out of reset for normal expanded mode when there is no internal memory at the reset vector address $FFFE–$FFFF. After reset in normal mode, the PCS stretch select bit in the CSCSTR register is set to provide one cycle of stretch so that slow memory devices can be used. In special test mode $\overline{\text{CSPROG}}$ is enabled without any stretch out of reset. Program chip select is fixed with address valid timing and is active low.

4.5.3 I/O chip select

The I/O chip select (CSIO) is programmable for a 4K byte size located at addresses $1000 to $1FFF, or 8K byte size located at addresses $0000 to $1FFF. Polarity of the active state is programmable for active high or active low. Clock stretching can be set from zero to three cycles. Refer to Section 4.5.4 for descriptions of bits IOEN, IOPL, IOCSA, and IOSZ.

4.5.4 CSCTL — Chip select control register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Chip select control (CSCTL)	$005B	IOEN	IOPL	IOCSA	IOSZ	GCSPR	PCSEN	PCSZA	PCSZB	0000 0100

IOEN — I/O chip select enable

1 (set) – CSIO is enabled and uses port H pin 4.

0 (clear) – CSIO is disabled and port H pin 4 is a general-purpose I/O pin.

IOPL — I/O chip select polarity select

1 (set) – CSIO is active high.

0 (clear) – CSIO is active low.

IOCSA — I/O chip select address valid

1 (set) – CSIO is valid during address valid time.

0 (clear) – CSIO is valid during E-clock high time.

IOSZ — I/O chip select size select

1 (set) – CSIO size is 8K at $0000–$1FFF.

0 (clear) – CSIO size is 4K at $1000–$1FFF.

GCSPR — General-purpose chip select priority (refer to Table 4-12)

1 (set) – General-purpose chip selects have priority over program chip select.

0 (clear) – Program chip select has priority over general-purpose chip selects.

PCSEN — Program chip select enable

1 (set) – CSPROG is enabled out of reset and uses port H pin 7.

0 (clear) – CSPROG is disabled and port H pin 7 is a general-purpose I/O pin.

PCSZA and PCSZB — Program chip select size (A or B)

Table 4-13 Program chip select size

PCSZA	PCSZB	Size (bytes)	Address range
0	0	64K	$0000 – $FFFF
0	1	32K	$8000 – $FFFF
1	0	16K	$C000 – $FFFF
1	1	8K	$E000 – $FFFF

4.5.5 General-purpose chip selects

The general-purpose chip selects are the most flexible and programmable and have the most control bits. Polarity of active state, E valid or address valid, size, and starting address are all programmable. Clock stretching can be set from zero to three cycles.

Each general-purpose chip select has two registers. One, the control register GPCSxC, determines the logical output required when an area of memory is selected and the range of memory over which the chip select is to be active. Each chip select can be programmed to become active whenever the CPU address enters a memory expansion window, regardless of the actual bank selected. This is known as following a window.

The second, the address register GPCSxA, allows the starting address of the chip select to be programmed. The bits in this register that are valid are determined by the size of the chip select range selected by the control register.

Refer to the descriptions of the two associated registers for starting address and control information.

In cases where one general-purpose chip select is programmed to drive the other general-purpose chip select or the program chip select, determine the priority from Table 4-14.

Table 4-14 General purpose chip select priority

Condition	Priority
GPCS1 drives GPCS2	GPCS1
GPCS1 drives PCS	GPCS1
GPCS2 drives PCS	GPCS2
GPCS1 and GPCS2 drive PCS	GPCS1

4.5.5.1 GPCS1A — General-purpose chip select 1 address register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Gen. purpose chip select 1 addr. (GPCS1A)	$005C	G1A18	G1A17	G1A16	G1A15	G1A14	G1A13	G1A12	G1A11	0000 0000

G1A[18:11] — General-purpose chip select 1 address

These bits select the starting address of general-purpose chip select 1 range. Refer to Table 4-15.

4.5.5.2 GPCS1C — General-purpose chip select 1 control register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Gen. purpose chip select 1 con. (GPCS1C)	$005D	G1DG2	G1DPC	G1POL	G1AV	G1SZA	G1SZB	G1SZC	G1SZD	0000 0000

G1DG2 — General-purpose chip select 1 drives general-purpose chip select 2

1 (set) – CSGP1 and CSGP2 are OR'ed and driven out CSGP2.

0 (clear) – CSGP1 does not affect CSGP2.

G1DPC — General-purpose chip select 1 drives program chip select

1 (set) – CSGP1 and $\overline{\text{CSPROG}}$ are OR'ed and driven out $\overline{\text{CSPROG}}$.

0 (clear) – CSGP1 does not affect $\overline{\text{CSPROG}}$.

G1POL — General-purpose chip select 1 polarity select

1 (set) – CSGP1 active high.

0 (clear) – CSGP1 active low.

G1AV — General-purpose chip select 1 address valid select

1 (set) – CSGP1 is valid during address valid time.

0 (clear) – CSGP1 is valid during E high time.

G1SZA–G1SZD — GP chip select 1 size

These bits select the size for general-purpose chip select 1. Refer to Table 4-15.

Table 4-15 General-purpose chip select 1 size control

G1SZx				Size (bytes)	Valid bits (MXGS1 = 0)	Valid bits (MXGS1 = 1)
A	B	C	D			
0	0	0	0	Disabled	None	None
0	0	0	1	2 K	G1A[15:11]	G1A[18:11]
0	0	1	0	4 K	G1A[15:12]	G1A[18:12]
0	0	1	1	8 K	G1A[15:13]	G1A[18:13]
0	1	0	0	16 K	G1A[15:14]	G1A[18:14]
0	1	0	1	32 K	A15	G1A[18:15]
0	1	1	0	64 K	None	G1A[18:16]
0	1	1	1	128 K	None	G1A[18:17]
1	0	0	0	256 K	None	G1A18
1	0	0	1	512 K	None	None
1	0	1	0	Follow window 1	None	None
1	0	1	1	Follow window 2	None	None
1100–1111				Default to 512 K	None	None

4.5.5.3 GPCS2A — General-purpose chip select 2 address register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Gen. purpose chip select 2 addr. (GPCS2A)	$005E	G2A18	G2A17	G2A16	G2A15	G2A14	G2A13	G2A12	G2A11	0000 0000

G2A[18:11] — General-purpose chip select 2 address

These bits select the starting address of general-purpose chip select 2 range. Refer to Table 4-16.

4.5.5.4 GPCS2C — General-purpose chip select 2 control register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Gen. purpose chip select 2 con. (GPCS2C)	$005F	0	G2DPC	G2POL	G2AV	G2SZA	G2SZB	G2SZC	G2SZD	0000 0000

Bit 7 — Not implemented; always reads zero.

G2DPC — General-purpose chip select 2 drives program chip select

1 (set) – CSGP2 and $\overline{\text{CSPROG}}$ are OR'ed and driven out the $\overline{\text{CSPROG}}$ pin.

0 (clear) – Does not affect program chip select.

G2POL — General-purpose chip select 2 polarity select

1 (set) – CSGP2 active high.

0 (clear) – CSGP2 active low.

G2AV — General-purpose chip select 2 address valid select

1 (set) – CSGP2 is valid during address valid time.

0 (clear) – CSGP2 is valid during E high time.

G2SZA–G2SZD — General-purpose chip select 2 size

These bits select the size for general-purpose chip select 2. Refer to Table 4-16.

Table 4-16 General-purpose chip select 2 size control

G2SZx				Size (bytes)	Valid bits (MXGS2 = 0)	Valid bits (MXGS2 = 1)
A	B	C	D			
0	0	0	0	Disabled	None	None
0	0	0	1	2K	G2A[15:11]	G2A[18:11]
0	0	1	0	4K	G2A[15:12]	G2A[18:12]
0	0	1	1	8K	G2A[15:13]	G2A[18:13]
0	1	0	0	16K	G2A[15:14]	G2A[18:14]
0	1	0	1	32K	G2A15	G2A[18:15]
0	1	1	0	64K	None	G2A[18:16]
0	1	1	1	128K	None	G2A[18:17]
1	0	0	0	256K	None	G2A18
1	0	0	1	512K	None	None
1	0	1	0	Follow Window 1	None	None
1	0	1	1	Follow Window 2	None	None
11100 – 1111				Default to 512K	None	None

4.5.6 One chip select driving another

The general-purpose chip selects can be programmed to drive the program chip select as well as each other. General-purpose chip select 1 drives general-purpose chip select 2 only. There are eight combinations of the bits G1DG2, G1DPC, and G2DPC. Although all possible combinations are allowed, some combinations cause operations which do not perform as one might expect. The results of all combinations are defined in the following table. The priorities defined in the previous sections still apply. The following table assumes that none of the chip select ranges overlap.

Table 4-17 One chip select driving another

G1DG 2	G1DPC	G2DP C	Program CS pin asserted when address is in:	General 2 CS pin asserted when address is in:	General 1 CS pin asserted when address is in:
0	0	0	A valid program area	A valid general 2 area	A valid general 1 area
0	0	1	A valid program or general 2 area	Never asserted	A valid general 1 area
0	1	0	A valid program or general 1 area	A valid general 2 area	Never asserted
0	1	1	A valid program or general 1 **or** 2 area	Never asserted	Never asserted
1	0	0	A valid program area	A valid general 2 **or** general 1 area	Never asserted
1	0	1	A valid program or general 2 area	Never asserted	A valid general 1 area
1	1	0	A valid program or general 1 area	A valid general 2 area	Never asserted
1	1	1	A valid program or general 1 **or** 2 area	Never asserted	Never asserted

4.5.7 Clock stretching

Each chip select has two bits that enable clock stretching from zero to three cycles. A clock stretch can be programmed to occur only during accesses to addresses in that chip select's address range. During the clock stretch period the E clock is held high for additional full periods and the bus remains in its normal state at the end of the E high time. Internally the clocks keep running so the integrity of the timers and baud rate generators is maintained.

4.5.7.1 CSCSTR — Chip select clock stretch register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Chip select clock stretch (CSCSTR)	$005A	IOSA	IOSB	GP1SA	GP1SB	GP2SA	GP2SB	PCSA	PCSB	0000 000x

Each of the following pairs of bits determines the clock stretch for one of the four chip selects.

IOSA, IOSB — CSIO stretch select

GP1SA, GP1SB — CSGP1 stretch select

GP2SA, GP2SB — CSGP2 stretch select

PCSA, PCSB — CSPROG stretch select

In normal modes (SMOD = 0), PCSB is set on reset to give a one cycle stretch. In special modes (SMOD = 1), PCSB is cleared on reset.

Bit [A: B]	Clock stretch
0 0	Disabled
0 1	1 cycle
1 0	2 cycles
1 1	3 cycles

Table 4-18 Chip select control parameter summary

CSIO (I/O chip select)		
Enable	IOEN in CSCTL —	1 = On, OFF at reset (0)
Valid	IOCSA in CSCTL —	1 = Address valid, 0 = E high
Polarity	IOPL in CSCTL —	1 = Active high, 0 = Active low
Size	IOSZ in CSCTL —	1 = 4K ($1000–$1FFF) 0 = 8K ($0000–$1FFF)
Start address	Fixed (see Size above)	
Stretch	IO1SA:IO1SB in CSCSTR — 0, 1, 2, or 3 E clocks	

CSPROG (program chip select)		
Enable	PSCEN in CSCTL — 1 = On, ON at reset	
Valid	Fixed (Address valid)	
Polarity	Fixed (Active low)	
Size	PCSZA:PCSZB — in CSCTL	0:0 = 64K ($0000–$FFFF) 0:1 = 32K ($8000–$FFFF) 1:0 = 16K ($C000–$FFFF) 1:1 = 8K ($E000–$FFFF)
Start address	Fixed (see Size above)	
Stretch	PCSA:PCSB in CSCSTR — 0, 1, 2, or 3 E clocks	
Priority	GCSPR in CSCTL —	1 = CSGPx above CSPROG 0 = CSPROG above CSGPx

CSGP1, CSGP2 (general purpose chip selects)	
Enable	Set size to 0K bytes to disable
Valid	Refer to GPCS1C / GPCS2C — Address valid or E high
Polarity	Refer to GPCS1C / GPCS2C — Active high or low
Size	Refer to GPCS1C / GPCS2C — 2K to 512K in nine steps, 0K bytes = disable, can also follow memory expansion window 1 or window 2
Start address	Refer to GPCS1A / GPCS2A
Stretch	Refer to CSCSTR — 0, 1, 2, or 3 E clocks
Other	G1DG2 in GPCS1C allows CSGP1 and CSGP2 to be logically OR'ed and driven out the CSGP2 pin. G1DPC in GPCS1C allows CSGP1 and CSPROG to be logically OR'ed and driven out the CSPROG pin. G2DPC in GPCS2C allows CSGP2 and CSPROG to be logically OR'ed and driven out the CSPROG pin. MXGS2 in MMSIZ allows CSGP2 to follow either 64K CPU addresses or 512K expansion addresses. MXGS1 in MMSIZ allows CSGP1 to follow either 64K CPU addresses or 512K expansion addresses.

4.6 EEPROM and CONFIG register

4.6.1 EEPROM

The 640-byte on-board EEPROM is initially located from $0D80 to $0FFF after reset in all modes. It can be mapped to any other 4K page by writing to the INIT2 register. The EEPROM is enabled by the EEON bit in the CONFIG register. Programming and erasing are controlled by the PPROG register.

Unlike information stored in ROM, data in the 640 bytes of EEPROM can be erased and reprogrammed under software control. Because programming and erasing operations use an on-chip charge pump driven by V_{DD}, a separate external power supply is not required.

An internal charge pump supplies the programming voltage. Seven bits in the block protect register (BPROT) prevent inadvertent writes to (or erases of) blocks of EEPROM, and the eighth bit enables the low voltage EEPROM protect circuit (see Section 4.3.2.6). The CSEL bit in the OPTION register selects an on-chip oscillator clock for programming and erasing the EEPROM while operating at frequencies below 1MHz.

In special modes there is one extra row of EEPROM, which is used for factory testing. Endurance and data retention specifications do not apply to these cells.

The erased state of each EEPROM byte is $FF.

4.6.1.1 PPROG — EEPROM programming control register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
EEPROM programming (PPROG)	$003B	ODD	EVEN	0	BYTE	ROW	ERASE	EELAT	EEPGM	0000 0000

Note: Writes to EEPROM addresses are inhibited while EEPGM is one. A write to a different EEPROM location is prevented while a program or erase operation is in progress.

ODD — Program odd rows in half of EEPROM (Test)

EVEN — Program even rows in half of EEPROM (Test)

If both ODD and EVEN are set to one then all odd and even rows in half of the EEPROM will be programmed with the same data, within one programming cycle.

BYTE — EEPROM byte erase mode

1 (set) – Erase only one byte of EEPROM.

0 (clear) – Row or bulk erase mode used.

This bit may be read or written at any time.

ROW — EEPROM row/bulk erase mode (only valid when BYTE = 0)

1 (set) – Erase only one 16 byte row of EEPROM.

0 (clear) – Erase all 640 bytes of EEPROM.

This byte can be read or written at any time.

Table 4-19 Erase mode selection

Byte	Row	Action
0	0	Bulk erase (all 640 bytes)
0	1	Row erase (16 bytes)
1	0	Byte erase
1	1	Byte erase

ERASE — Erase/normal control for EEPROM

1 (set) – Erase mode.

0 (clear) – Normal read or program mode.

This byte can be read or written at any time.

EELAT — EEPROM latch control

1 (set) – EEPROM address and data bus set up for programming or erasing.

0 (clear) – EEPROM address and data bus set up for normal reads.

When the EELAT bit is cleared, the EEPROM can be read as if it were a ROM. The block protect register has no effect during reads. This bit can be read and written at any time.

EEPGM — EEPROM program command

1 (set) – Program or erase voltage switched on to EEPROM array.

0 (clear) – Program or erase voltage switched off to EEPROM array.

This bit can be read at any time but can only be written if EELAT = 1.

Note: If EELAT = 0 (normal operation) then EEPGM = 0 (programming voltage disconnected).

During EEPROM programming, the ROW and BYTE bits of PPROG are not used. If the frequency of the E clock is 1MHz or less, set the CSEL bit in the OPTION register. Remember that the EEPROM must be erased by a separate erase operation before programming. The following

example of how to program an EEPROM byte assumes that the appropriate bits in BPROT have been cleared.

```
PROG    LDAB    #$02    EELAT=1
        STAB    $003B   Set EELAT bit
        STAA    $0D80   Store data to EEPROM address
        LDAB    #$03    EELAT=EEPGM=1
        STAB    $003B   Turn on programming voltage
        JSR     DLY10   Delay tEEPROG
        CLR     $003B   Turn off high voltage and set to READ mode
```


4.6.1.2 EEPROM bulk erase

To erase the EEPROM, ensure that the appropriate bits in the BPROT register are cleared, then complete the following steps using the PPROG register:

1) Write to PPROG with the ERASE, EELAT and appropriate BYTE and ROW bits set.
2) Write to the appropriate EEPROM address with any data. Row erase only requires a write to any location in the row. Bulk erase is accomplished by writing to any location in the array.
3) Write to PPROG with ERASE, EELAT, EEPGM and the appropriate BYTE and ROW bits set.
4) Delay for time t_{EEPROG} (See Section A.6).
5) Clear the EEPGM bit in PPROG to turn off the high voltage.
6) Clear the PPROG register to reconfigure the EEPROM address and data buses for normal operation.

The following is an example of how to bulk erase the 640-byte EEPROM. The CONFIG register is not affected in this example.

```
BULKE   LDAB    #$06    EELAT=ERASE=1
        STAB    $003B   Set EELAT bit
        STAA    $0D80   Store data to any EEPROM address
        LDAB    #$07    EELAT=ERASE=EEPGM=1
        STAB    $003B   Turn on programming voltage
        JSR     DLY10   Delay tEEPROG
        CLR     $003B   Turn off high voltage and set to READ mode
```

4.6.1.3 EEPROM row erase

The following example shows how to perform a fast erase of 16 bytes of EEPROM:

```
ROWE    LDAB    #$0E    ROW=ERASE=EELAT=1
        STAB    $003B   Set to ROW erase mode
```

```
        STAB    0,X     Write any data to any address in ROW
        LDAB    #$0F    ROW=ERASE=EELAT=EEPGM=1
        STAB    $003B   Turn on high voltage
        JSR     DLY10   Delay tEEPROG
        CLR     $003B   Turn off high voltage and set to READ mode
```

4.6.1.4 EEPROM byte erase

The following is an example of how to erase a single byte of EEPROM:

```
BYTEE   LDAB    #$16    BYTE=ERASE=EELAT=1
        STAB    $003B   Set to BYTE erase mode
        STAB    0,X     Write any data to address to be erased
        LDAB    #$17    BYTE=ERASE=EELAT=EEPGM=1
        STAB    $003B   Turn on high voltage
        JSR     DLY10   Delay tEEPROG
        CLR     $003B   Turn off high voltage and set to READ mode
```

4.6.2 CONFIG register programming

Because the CONFIG register is implemented with EEPROM cells, use EEPROM procedures to erase and program this register. The procedure for programming is the same as for programming a byte in the EEPROM array, except that the CONFIG register address is used. CONFIG can be programmed or erased (including byte erase) while the MCU is operating in any mode, provided that PTCON in BPROT is clear. To change the value in the CONFIG register, complete the following procedure. Do not initiate a reset until the procedure is complete.

1) Erase the CONFIG register.
2) Program the new value to the CONFIG address.
3) Initiate reset.

CONFIG — System configuration register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Configuration control (CONFIG)	$003F	1	1	CLKX	PAREN	NOSEC	NOCOP	1	EEON	11xx xx1x

For a description of the bits contained in the CONFIG register refer to Section 4.3.2.1.

CONFIG is made up of EEPROM cells and static working latches. The operation of the MCU is controlled directly by these latches and not the EEPROM byte. When programming the CONFIG register, the EEPROM byte is accessed. When the CONFIG register is read, the static latches are accessed.

These bits can be read at any time. The value read is the one latched into the register from the EEPROM cells during the last reset sequence. A new value programmed into this register is not readable until after a subsequent reset sequence.

Bits in CONFIG can be written at any time if SMOD = 1 (bootstrap or special test mode). If SMOD = 0 (single chip or expanded mode), these bits can only be written using the EEPROM programming sequence, and none of the bits is readable or active until latched via the next reset.

4.6.3 RAM and EEPROM security

The optional security feature protects the contents of EEPROM and RAM from unauthorized access. Data, codes, keys, a program, or a key portion of a program, can be protected against access. To accomplish this, the protection mechanism prevents operation of the device in special test mode. Only resident programs have unlimited access to the internal EEPROM and RAM and can read, write, or transfer the contents of these memories. To maintain RAM and EEPROM security, access to external addresses should be restricted to data read or write. Program execution should not point from the internal resources to the external memory map.

Note: A mask option on the MC68HC11KW1 determines whether or not the security feature is made available. If the feature is available, then the secure mode can be invoked by programming the NOSEC bit to zero. Otherwise, the NOSEC bit is permanently set to one, disabling security.

If the security feature is present and enabled and bootstrap mode is selected, then the following sequence is performed by the bootstrap program:

1) Output $FF on the SCI.
2) Turn block protect off. Clear BPROT register.
3) If EEPROM is enabled, erase it all.
4) Verify that the EEPROM is erased; if not, begin sequence again.
5) Write $FF to every RAM byte.
6) Erase the CONFIG register.

If all the above operations are successful, the bootloading process continues as if the device has not been secured.

CONFIG — System configuration register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Configuration control (CONFIG)	$003F	1	1	CLKX	PAREN	NOSEC	NOCOP	1	EEON	11xx xx1x

For a description of the other bits contained in the CONFIG register refer to Section 4.3.2.1.

NOSEC — EEPROM security disabled

1 (set) – Disable security.

0 (clear) – Enable security.

With security enabled, selection of special test mode is prevented; single chip and user expanded modes may be accessed. If the MODA and MODB pins are configured for special test mode, the part will start in bootstrap mode.

5
RESETS AND INTERRUPTS

Resets and interrupt operations load the program counter with a vector that points to a new location from which instructions are to be fetched. A reset immediately stops execution of the current instruction and forces the program counter to a known starting address. Internal registers and control bits are initialized so that the MCU can resume executing instructions. An interrupt temporarily suspends normal program execution whilst an interrupt service routine is being executed. After an interrupt has been serviced, the main program resumes as if there had been no interruption.

5.1 Resets

There are four possible sources of reset. Power-on reset (POR) and external reset share the normal reset vector. The computer operating properly (COP) reset and the clock monitor reset each has its own vector.

5.1.1 Power-on reset

A positive transition on VDD generates a power-on reset (POR), which is used only for power-up conditions. POR cannot be used to detect drops in power supply voltages. A delay is imposed which allows the clock generator to stabilize after the oscillator becomes active. If $\overline{\text{RESET}}$ is at logical zero at the end of the delay time, the CPU remains in the reset condition until $\overline{\text{RESET}}$ goes to logical one.

It is important to protect the MCU during power transitions. Most M68HC11 systems need an external circuit that holds the $\overline{\text{RESET}}$ pin low whenever V_{DD} is below the minimum operating level. This 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 power on. Refer to Figure 2-3.

5.1.2 External reset ($\overline{\text{RESET}}$)

The CPU distinguishes between internal and external reset conditions by sensing whether the reset pin rises to a logic one in less than four 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 eight E clock cycles, then released. Four 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. It is not advisable to connect an external resistor capacitor (RC) power-up delay circuit to the reset pin of M68HC11 devices because the circuit charge time constant can cause the device to misinterpret the type of reset that occurred. To guarantee recognition of an external reset, the $\overline{\text{RESET}}$ pin should be held low for at least 16 clock cycles.

5.1.3 COP reset

The MCU includes a COP system to help protect against software failures. When the COP is enabled, the software is responsible for keeping a free-running watchdog timer from timing out. When the software is no longer being executed in the intended sequence, a system reset is initiated.

The state of the NOCOP bit in the CONFIG register determines whether the COP system is enabled or disabled. To change the enable status of the COP system, change the contents of the CONFIG register and then perform a system reset. In the special test and bootstrap operating modes, the COP system is initially inhibited by the disable resets (DISR) control bit in the TEST1 register. The DISR bit can subsequently be written to zero to enable COP resets.

The COP timer rate control bits, CR[1:0], in the OPTION register determine the COP timeout period. The system E clock is divided by 2^{15} and then further scaled by the factor shown in Table 5-1. After reset, bits CR[1:0] are zero, which selects the shortest timeout period. In normal operating modes, these bits can only be written once, within 64 bus cycles after reset.

Table 5-1 COP timer rate select

CR[1:0]	Divide E/2^{15} by	EXTAL = 16 MHz: timeout (1)
0 0	1	8.192 ms
0 1	4	32.768 ms
1 0	16	131.072 ms
1 1	64	524.288 ms
	E =	**4 MHz**

(1) The timeout period has a tolerance of –0/+one cycle of the E/2^{15} clock due to the asynchronous implementation of the COP circuitry. For example, with E = 4 MHz, the uncertainty is –0/+8.192 ms. See also the ***M68HC11 Reference Manual, (M68HC11RM/AD)***.

5.1.3.1 COPRST — Arm/reset COP timer circuitry register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
COP timer arm/reset (COPRST)	$003A	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	not affected

Complete the following reset sequence to service the COP timer. Write $55 to COPRST to arm the COP timer clearing mechanism. Then write $AA to COPRST to clear the COP timer. Executing instructions between these two steps is possible as long as both steps are completed in the correct sequence before the timer times out.

5.1.4 Clock monitor reset

The clock monitor circuit is based on an internal 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 or disabled by the CME control bit in the OPTION register. The presence of a timeout is determined by the RC delay, which allows the clock monitor to operate without any MCU clocks.

Clock monitor is used as a backup for the COP system. Because the COP needs a clock to function, it is disabled when the clocks stop. Therefore, the clock monitor system can detect clock failures not detected by the COP system.

Semiconductor wafer processing causes variations of the RC timeout values between individual devices. An E clock frequency below 10 kHz is detected as a clock monitor error. An E clock frequency of 200 kHz or more prevents clock monitor errors. Use of the clock monitor function when the E clock is below 200 kHz is not recommended.

Special considerations are needed when a STOP instruction is executed and the clock monitor is enabled. Because the STOP function causes the clocks to be halted, the clock monitor function generates a reset sequence if it is enabled at the time the STOP mode was initiated. Before executing a STOP instruction, clear the CME bit in the OPTION register to zero to disable the clock monitor. After recovery from STOP, set the CME bit to logic one to enable the clock monitor.

5.1.5 OPTION — System configuration options register 1

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
System config. options 1 (OPTION)	$0039	ADPU	CSEL	IRQE	DLY	CME	FCME	CR1	CR0	0001 0000

The special-purpose OPTION register sets internal system configuration options during initialization. The time protected control bits (IRQE, DLY, FCME and CR[1:0]) can be written to only once in the first 64 cycles after a reset and then they become read-only bits. This minimizes the possibility of any accidental changes to the system configuration. They may be written at any time in special modes.

ADPU — A/D power-up (refer to Section 10)

1 (set) – A/D system power enabled.

0 (clear) – A/D system disabled, to reduce supply current.

CSEL — Clock select (refer to Section 10)

1 (set) – A/D and EEPROM use internal RC clock (about 1.5 MHz).

0 (clear) – A/D and EEPROM use system E clock (must be at least 1 MHz).

IRQE — Configure $\overline{\text{IRQ}}$ for falling-edge-sensitive operation (refer to Section 4)

1 (set) – Falling-edge-sensitive operation.

0 (clear) – Low-level-sensitive operation.

DLY — Enable oscillator start-up delay (refer to Section 4)

1 (set) – A stabilization delay is imposed as the MCU is started up from STOP mode (or from power-on reset).

0 (clear) – The oscillator start-up delay is bypassed and the MCU resumes processing within about four bus cycles. A stable external oscillator is required if this option is selected.

Note: Because DLY is set on reset, a delay is always imposed as the MCU is started up from power-on reset.

CME — Clock monitor enable

1 (set) – Clock monitor enabled.

0 (clear) – Clock monitor disabled.

This control bit can be read or written at any time and controls whether or not the internal clock monitor circuit triggers a reset sequence when the system clock is slow or absent. When it is clear, the clock monitor circuit is disabled, and when it is set, the clock monitor circuit is enabled. Reset clears the CME bit.

In order to use both STOP and clock monitor, the CME bit should be cleared before executing STOP, then set again after recovering from STOP.

FCME — Force clock monitor enable

1 (set) – Clock monitor enabled; cannot be disabled until next reset.

0 (clear) – Clock monitor follows the state of the CME bit.

When FCME is set, slow or stopped clocks will cause a clock failure reset sequence. To utilize STOP mode, FCME should always be cleared.

CR[1:0] — COP timer rate select bits

The internal E clock is first divided by 2^{15} before it enters the COP watchdog system. These control bits determine a scaling factor for the watchdog timer period. See Table 5-1.

5.1.6 CONFIG — Configuration control register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Configuration control (CONFIG)	$003F	1	1	CLKX	PAREN	NOSEC	NOCOP	1	EEON	11xx 1x1x

Included in CONFIG are bits which control the presence of EEPROM in the memory map and enable the COP watchdog system.

CONFIG is made up of EEPROM cells and static working latches. The operation of the MCU is controlled directly by these latches and not the EEPROM byte. When programming the CONFIG register, the EEPROM byte is accessed. When the CONFIG register is read, the static latches are accessed.

These bits can be read at any time. The value read is the one latched into the register from the EEPROM cells during the last reset sequence. A new value programmed into this register is not readable until after a subsequent reset sequence.

Bits in CONFIG can be written at any time if SMOD = 1 (bootstrap or special test mode). If SMOD = 0 (single chip or expanded mode), these bits can only be written using the EEPROM programming sequence, and none of the bits are readable or active until latched via the next reset.

Bits [7, 6, 1] — Not implemented; always read one.

CLKX — X clock enable (refer to Section 4)

1 (set) – XCLK driven out on the XOUT pin.

0 (clear) – XOUT pin disabled.

PAREN — Pull-up assignment register enable (refer to Section 6)

1 (set) – PPAR register enabled; pull-ups can be enabled using PPAR.

0 (clear) – PPAR register disabled; all pull-ups disabled.

NOSEC — Security disable (refer to Section 4)

1 (set) – Disable security.

0 (clear) – Enable security.

NOCOP — COP system disable

1 (set) – COP system disabled.

0 (clear) – COP system enabled (forces reset on timeout).

EEON — EEPROM enable (refer to Section 4)

1 (set) – EEPROM included in the memory map.

0 (clear) – EEPROM excluded from the memory map.

5.2 Effects of reset

When a reset condition is recognized, the internal registers and control bits are forced to an initial state. Depending on the cause of the reset and the operating mode, the reset vector can be fetched from any of six possible locations, as shown in Table 5-2.

Table 5-2 Reset cause, reset vector and operating mode

Cause of reset	Normal mode vector	Special test or bootstrap
POR or $\overline{\text{RESET}}$ pin	$FFFE, $FFFF	$BFFE, $BFFF
Clock monitor failure	$FFFC, $FFFD	$BFFC, $BFFD
COP watchdog timeout	$FFFA, $FFFB	$BFFA, $BFFB

These initial states then control on-chip peripheral systems to force them to known start-up states, as described in the following paragraphs.

5.2.1 Central processing unit

After reset, the CPU fetches the restart vector from the appropriate address during the first three cycles, and begins executing instructions. The stack pointer and other CPU registers are indeterminate immediately after reset; however, the X and I interrupt mask bits in the condition code register (CCR) are set to mask any interrupt requests. Also, the S-bit in the CCR is set to inhibit the STOP mode.

5.2.2 Memory map

After reset, the INIT register is initialized to $00, putting the 768 bytes of RAM at locations $00A0–$039F, and the control registers at locations $0000–$009F. The INIT2 register puts EEPROM at locations $0D80–$0FFF.

5.2.3 Parallel I/O

When a reset occurs in expanded operating modes, port B, C, and F pins used for parallel I/O are dedicated to the expansion bus. If a reset occurs during a single chip operating mode, all ports are configured as general purpose high-impedance inputs.

Note: Do not confuse pin function with the electrical state of the pin at reset. All general-purpose I/O pins configured as inputs at reset are in a high-impedance state. Port data registers reflect the port's functional state at reset. The pin function is mode dependent.

5.2.4 Timer 1

During reset, the Timer 1 system is initialized to a count of $0000. The prescaler bits for Timer 1 are cleared, and all output compare registers are initialized to $FFFF. All input capture registers are indeterminate after reset. The output compare 1 mask (OC1M) register is cleared so that successful OC1 compares do not affect any I/O pins. The other four output compares are configured so that they do not affect any I/O pins on successful compares. All input capture edge-detector circuits are configured for capture disabled operation. The timer overflow interrupt flags and all eight timer function interrupt flags are cleared. All nine timer interrupts are disabled because their mask bits have been cleared.

The I4/O5 bit in the PACTL register is cleared to configure the I4/O5 function as OC5; however, the OM5:OL5 control bits in the TCTL1 register are clear so OC5 does not control the PA3 pin.

5.2.5 Timers 2 and 3

During reset, each of these timer systems is initialized to a count of $0000. The ECEB, ECEA and prescaler bits are cleared so that the timers are driven by the internal E clock. The output compare registers are initialized to $FFFF. The interrupt flag registers (T2FLG and T3FLG) are cleared, along with the interrupt mask registers (T2MSK and T3MSK), disabling all interrupts.

For each timer, the I1/O4 bit is clear to configure C4 as OC4, however, the OM4:OL4 bits are clear so that OC4 does not affect the corresponding port pin.

5.2.6 Real-time interrupt (RTI)

The real-time interrupt flag (RTIF) is cleared and automatic hardware interrupts are masked. The rate control bits are cleared after reset and can be initialized by software before the real-time interrupt (RTI) system is used.

5.2.7 Pulse accumulator

The pulse accumulator system is disabled at reset so that the pulse accumulator input (PAI) pin defaults to being a general-purpose input pin.

5.2.8 Computer operating properly (COP)

The COP watchdog system is enabled if the NOCOP control bit in the CONFIG register is cleared, and disabled if NOCOP is set. The COP rate is set for the shortest duration timeout.

5.2.9 Serial communications interface (SCI)

The reset condition of the SCI system is independent of the operating mode. At reset, the SCI baud rate control register is initialized to $0004. All transmit and receive interrupts are masked and both the transmitter and receiver are disabled so the port pins default to being general purpose I/O lines. The SCI frame format is initialized to an 8-bit character size. The send break and receiver wake-up functions are disabled. The TDRE and TC status bits in the SCI status register are both set, indicating that there is no transmit data in either the transmit data register or the transmit serial shift register. The RDRF, IDLE, OR, NF, FE, PF, and RAF receive-related status bits are cleared.

5.2.10 Serial peripheral interface (SPI)

The SPI system is disabled by reset. Its associated port pins default to being general-purpose I/O lines.

5.2.11 Analog-to-digital converter

The A/D converter configuration is indeterminate after reset. The ADPU bit is cleared by reset, which disables the A/D system.

5.2.12 System

The EEPROM programming controls are disabled, so the memory system is configured for normal read operation. PSEL[4:0] are initialized with the binary value%00110, 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). The RBOOT, SMOD, and MDA bits in the HPRIO register reflect the status of the MODB and MODA inputs at the rising edge of reset. The DLY control bit is set to specify that an oscillator start-up delay is imposed upon recovery from STOP mode or power-on reset. The clock monitor system is disabled because CME and FCME are cleared.

5.3 Reset and interrupt priority

Resets and interrupts have a hardware priority that determines which reset or interrupt is serviced first when simultaneous requests occur. Any maskable interrupt can be given priority over other maskable interrupts.

The first six interrupt sources are not maskable by the I-bit in the CCR. The priority arrangement for these sources is fixed and is as follows:

1) POR or $\overline{\text{RESET}}$ pin
2) Clock monitor reset
3) COP watchdog reset
4) $\overline{\text{XIRQ}}$ interrupt
– Illegal opcode interrupt — see Section 5.4.3 for details of handling
– Software interrupt (SWI) — see Section 5.4.4 for details of handling

The maskable interrupt sources have the following priority arrangement:

5) $\overline{\text{IRQ}}$
6) Real-time interrupt
7) Timer 1 input capture 1

8) Timer 1 input capture 2
9) Timer 1 input capture 3
10) Timer 1 output compare 1
11) Timer 1 output compare 2
12) Timer 1 output compare 3
13) Timer 1 output compare 4
14) Timer 1 input capture 4/output compare 5
15) Timer 2 output compare 1, 2, 3
16) Timer 2 input capture 1/output compare 4
17) Timer 1 overflow
18) Timer 2 overflow
19) Pulse accumulator overflow
20) Pulse accumulator input edge
21) Timer 3 capture/compare
22) Timer 3 overflow
23) SPI transfer complete
24) SCI system

Any one of these maskable interrupts can be assigned the highest maskable interrupt priority by writing the appropriate value to the PSEL bits in the HPRIO register. Otherwise, the priority arrangement remains 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. To avoid race conditions, HPRIO can only be written while I-bit interrupts are inhibited.

5.3.1 HPRIO — Highest priority I-bit interrupt and misc. register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Highest priority interrupt (HPRIO)	$003C	RBOOT	SMOD	MDA	PSEL4	PSEL3	PSEL2	PSEL1	PSEL0	xxx0 0110

RBOOT, SMOD, and MDA bits depend on power-up initialization mode and can only be written in special modes when SMOD = 1. Refer to Table 4-4.

RBOOT — Read bootstrap ROM (refer to Section 4)

1 (set) – Bootloader ROM enabled, at $BE40–$BFFF.

0 (clear) – Bootloader ROM disabled and not in map.

SMOD — Special mode select (refer to Section 4)

1 (set) – Special mode variation in effect.

0 (clear) – Normal mode variation in effect.

MDA — Mode select A (refer to Section 4)

1 (set) – Normal expanded or special test mode in effect.

0 (clear) – Normal single chip or special bootstrap mode in effect.

PSEL[4:0] — Priority select bits

These bits select one interrupt source to be elevated above all other I-bit-related sources and can be written to only while the I-bit in the CCR is set (interrupts disabled). See Table 5-3.

Table 5-3 Highest priority interrupt selection

PSELx					Interrupt source promoted
4	3	2	1	0	
0	0	0	X	X	Reserved (default to $\overline{IRQ}$)
0	0	1	0	0	Reserved (default to $\overline{IRQ}$)
0	0	1	0	1	Reserved (default to $\overline{IRQ}$)
0	0	1	1	0	$\overline{IRQ}$ (external pin)
0	0	1	1	1	Real-time interrupt
0	1	0	0	0	Timer 1 input capture 1
0	1	0	0	1	Timer 1 input capture 2
0	1	0	1	0	Timer 1 input capture 3
0	1	0	1	1	Timer 1 output compare 1
0	1	1	0	0	Timer 1 output compare 2
0	1	1	0	1	Timer 1 output compare 3
0	1	1	1	0	Timer 1 output compare 4
0	1	1	1	1	Timer 1 output compare 5/input capture 4
1	0	1	0	1	Timer 2 output compare 1, 2, 3
1	0	1	1	0	Timer 2 input capture1/output compare 4
1	0	0	0	0	Timer 1 overflow
1	0	1	1	1	Timer 2 overflow
1	0	0	0	1	Pulse accumulator overflow
1	0	0	1	0	Pulse accumulator input edge
1	1	0	0	0	Timer 3 input capture/output compare
1	1	0	0	1	Timer 3 overflow
1	0	0	1	1	SPI serial transfer complete
1	0	1	0	0	SCI serial system
1	1	0	1	X	Reserved (default to $\overline{IRQ}$)
1	1	1	X	X	Reserved (default to $\overline{IRQ}$)

Table 5-4 Interrupt and reset vector assignments

Vector address	Interrupt source	CCR mask bit	Local mask
FFC0, C1 – FFCA, CB	• Reserved	—	—
FFCC, FFCD	•Timer 3 overflow	I	TO3I
FFCE, FFCF	•Timer 3 input capture 1/output compare 4 •Timer 3 output compare 1 •TImer 3 output compare 2 •Timer 3 output compare 3	I	C4I OC1I OC2I OC3I
FFD0, FFD1	Timer 2 overflow	I	TO2I
FFD2, FFD3	Timer 2 input capture 1/output compare 4	I	C4I
FFD4, D5	•Timer 2 output compare 1 •Timer 2 output compare 2 •Timer 2 output compare 3	I	OC1I OC2I OC3I
FFD6, D7	• SCI receive data register full • SCI receiver overrun • SCI transmit data register empty • SCI transmit complete • SCI idle line detect	I	RIE RIE TIE TCIE ILIE
FFD8, D9	SPI serial transfer complete	I	SPIE
FFDA, DB	Pulse accumulator input edge	I	PAII
FFDC, DD	Pulse accumulator overflow	I	PAOVI
FFDE, DF	Timer 1 overflow	I	TOI
FFE0, E1	Timer 1 input capture 4/output compare 5	I	I4/O5I
FFE2, E3	Timer 1 output compare 4	I	OC4I
FFE4, E5	Timer 1 output compare 3	I	OC3I
FFE6, E7	Timer 1 output compare 2	I	OC2I
FFE8, E9	Timer 1 output compare 1	I	OC1I
FFEA, EB	Timer 1 input capture 3	I	IC3I
FFEC, ED	Timer 1 input capture 2	I	IC2I
FFEE, EF	Timer 1 input capture 1	I	IC1I
FFF0, F1	Real-time interrupt	I	RTII
FFF2, F3	$\overline{\text{IRQ}}$ pin	I	None
FFF4, F5	$\overline{\text{XIRQ}}$ pin	X	None
FFF6, F7	Software interrupt	None	None
FFF8, F9	Illegal opcode trap	None	None
FFFA, FB	COP failure	None	NOCOP
FFFC, FD	Clock monitor fail	None	CME
FFFE, FF	$\overline{\text{RESET}}$	None	None

5.4 Interrupts

Excluding reset type interrupts, the MC68HC11KW1 has 23 interrupt vectors that support 32 interrupt sources. The 20 maskable interrupts are generated by on-chip peripheral systems. These interrupts are recognized when the global interrupt mask bit (I) in the condition code register (CCR) is clear. The three nonmaskable interrupt sources are illegal opcode trap, software interrupt, and $\overline{\text{XIRQ}}$ pin. Refer to Table 5-4, which shows the interrupt sources and vector assignments for each source.

For some interrupt sources, such as the SCI interrupts, the flags are automatically cleared during the normal course of responding to the interrupt requests. For example, the RDRF flag in the SCI system is cleared by the automatic clearing mechanism consisting of a read of the SCI status register while RDRF is set, followed by a read of the SCI data register. The normal response to an RDRF interrupt request would be to read the SCI status register to check for receive errors, then to read the received data from the SCI data register. These two steps satisfy the automatic clearing mechanism without requiring any special instructions.

5.4.1 Interrupt recognition and register stacking

An interrupt can be recognized at any time after it is enabled by its local mask, if any, and by the global mask bit in the CCR. Once an interrupt source 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 current instruction. When the CPU begins to service an interrupt, the contents of the CPU registers are pushed onto the stack in the order shown in Table 5-5. After the CCR value is stacked, the I-bit and the X-bit, if $\overline{\text{XIRQ}}$ is pending, are set to inhibit further interrupts. The interrupt vector for the highest priority pending source is fetched, and execution continues at the address specified by the vector. At the end of the interrupt service routine, the return from interrupt instruction is executed and the saved registers are pulled from the stack in reverse order so that normal program execution can resume. Refer to Section 3 for further information.

Table 5-5 Stacking order on entry to interrupts

Memory location	CPU registers
SP	PCL
SP – 1	PCH
SP – 2	IYL
SP – 3	IYH
SP – 4	IXL
SP – 5	IXH
SP – 6	ACCA
SP – 7	ACCB
SP – 8	CCR

5.4.2 Nonmaskable interrupt request ($\overline{\text{XIRQ}}$)

Nonmaskable interrupts are useful because they can always interrupt CPU operations. The most common use for such an interrupt is for serious system problems, such as program runaway or power failure. The $\overline{\text{XIRQ}}$ input is an updated version of the $\overline{\text{NMI}}$ (nonmaskable interrupt) input of earlier MCUs.

Upon reset, both the X-bit and I-bit of the CCR are set to inhibit all maskable interrupts and $\overline{\text{XIRQ}}$. After minimum system initialization, software can clear the X-bit by a TAP instruction, enabling $\overline{\text{XIRQ}}$ interrupts. Thereafter, software cannot set the X-bit. Thus, an $\overline{\text{XIRQ}}$ interrupt is a nonmaskable interrupt. Because the operation of the I-bit-related interrupt structure has no effect on the X-bit, the internal $\overline{\text{XIRQ}}$ pin remains unmasked. In the interrupt priority logic, the $\overline{\text{XIRQ}}$ interrupt has a higher priority than any source that is maskable by the I-bit. All I-bit-related interrupts operate normally with their own priority relationship.

When an I-bit-related interrupt occurs, the I-bit is automatically set by hardware after stacking the CCR byte. The X-bit is not affected. When an X-bit-related interrupt occurs, both the X and I bits are automatically set by hardware after stacking the CCR. A return from interrupt instruction restores the X and I bits to their pre-interrupt request state.

5.4.3 Illegal opcode trap

Because not all possible opcodes or opcode sequences are defined, the MCU includes an illegal opcode detection circuit, which generates an interrupt request. When an illegal opcode is detected and the interrupt is recognized, the current value of the program counter is stacked. After interrupt service is complete, the user should reinitialize the stack pointer to ensure that repeated execution of illegal opcodes does not cause stack underflow. Left uninitialized, the illegal opcode vector can point to a memory location that contains an illegal opcode. This condition causes an infinite loop that causes stack underflow. The stack grows until the system crashes.

The illegal opcode trap mechanism works for all unimplemented opcodes on all four opcode map pages. The address stacked as the return address for the illegal opcode interrupt is the address of the first byte of the illegal opcode. Otherwise, it would be almost impossible to determine whether the illegal opcode had been one or two bytes. The stacked return address can be used as a pointer to the illegal opcode, so that the illegal opcode service routine can evaluate the offending opcode.

5.4.4 Software interrupt

SWI is an instruction, and thus cannot be interrupted until complete. SWI is not inhibited by the global mask bits in the CCR. Because execution of SWI sets the I mask bit, once an SWI interrupt begins, other interrupts are inhibited until SWI is complete, or until user software clears the I bit in the CCR.

5.4.5 Maskable interrupts

The maskable interrupt structure of the MCU can be extended to include additional external interrupt sources through the $\overline{IRQ}$ pin. The default configuration of this pin is a low-level sensitive wired-OR network. When an event triggers an interrupt, a software accessible interrupt flag is set. When enabled, this flag causes a constant request for interrupt service. After the flag is cleared, the service request is released.

5.4.6 Reset and interrupt processing

The following flow diagrams illustrate the reset and interrupt process. Figure 5-1 and Figure 5-2 illustrate how the CPU begins from a reset and how interrupt detection relates to normal opcode fetches. Figure 5-3 to Figure 5-4 provide an expanded version of a block in Figure 5-1 and illustrate interrupt priorities. Figure 5-6 shows the resolution of interrupt sources within the SCI subsystem.

5.5 Low power operation

Both STOP and WAIT suspend CPU operation until a reset or interrupt occurs. The WAIT condition suspends processing and reduces power consumption to an intermediate level. The STOP condition turns off all on-chip clocks and reduces power consumption to an absolute minimum while retaining the contents of all bytes of the RAM.

5.5.1 WAIT

The WAI opcode places the MCU in the WAIT condition, during which the CPU registers are stacked and CPU processing is suspended until a qualified interrupt is detected. The interrupt can be an external $\overline{IRQ}$, an $\overline{XIRQ}$, or any of the internally generated interrupts, such as the timer or serial interrupts. The on-chip crystal oscillator remains active throughout the WAIT stand-by period.

The reduction of power in the WAIT condition depends on how many internal clock signals driving on-chip peripheral functions can be shut down. The CPU is always shut down during WAIT. While in the wait state, the address/data bus repeatedly runs read cycles to the address where the CCR contents were stacked. The MCU leaves the wait state when it senses any interrupt that has not been masked.

The free-running Timer 1 system is stopped only if the I-bit is set and the COP system is disabled by NOCOP being set. Timers 2 and 3 can be stopped under the control of bits in the TCTL4 and TCTL6 registers, respectively. Several other systems can also be in a reduced power consumption state depending on the state of software-controlled configuration control bits. Power consumption by the analog-to-digital (A/D) converter is not affected significantly by the WAIT condition.

However, the A/D converter current can be reduced by writing the ADPU bit to zero and halting the RC clock (CSEL cleared). If the reference voltages V_{RH}, V_{RL} are supplied then the interval resistor chain still consumes current. For 5V V_{RH} - V_{RL}, a worst case current consumption of 260μA is specified for the A/D converter. The SPI system is enabled or disabled by the SPE control bit. The SCI transmitter is enabled or disabled by the TE bit, and the SCI receiver is enabled or disabled by the RE bit (lowest power consumption is achieved when RE=TE=0). Power consumption is reduced if all the PWM enable bits (PWEN[4:1]) are cleared, thereby disabling every PWM channel. Therefore the power consumption in WAIT is dependent on the particular application.

5.5.2 STOP

Executing the STOP instruction while the S-bit in the CCR is clear places the MCU in the STOP condition. If the S-bit is set, the STOP opcode is treated as a no-op (NOP). The STOP condition offers minimum power consumption because all clocks, including the crystal oscillator, are stopped while in this mode. To exit STOP and resume normal processing, a logic low level must be applied to one of the external interrupts ($\overline{IRQ}$ or $\overline{XIRQ}$) or to the $\overline{RESET}$ pin. A pending edge-triggered $\overline{IRQ}$ can also bring the CPU out of STOP.

Because all clocks are stopped in this mode, all internal peripheral functions also stop. The data in the internal RAM is retained as long as V_{DD} power is maintained. The CPU state and I/O pin levels are static and are unchanged by STOP. Therefore, when an interrupt comes to restart the system, the MCU resumes processing as if there were no interruption. If reset is used to restart the system a normal reset sequence results where all I/O pins and functions are also restored to their initial states.

To use the $\overline{IRQ}$ pin as a means of recovering from STOP, the I-bit in the CCR must be clear ($\overline{IRQ}$ not masked). The $\overline{XIRQ}$ pin can be used to wake up the MCU from STOP regardless of the state of the X-bit in the CCR, although the recovery sequence depends on the state of the X-bit. If X is clear ($\overline{XIRQ}$ not masked), the MCU starts up, beginning with the stacking sequence leading to normal service of the $\overline{XIRQ}$ request. If X is set ($\overline{XIRQ}$ masked or inhibited), then processing continues with the instruction that immediately follows the STOP instruction, and no $\overline{XIRQ}$ interrupt service is requested or pending.

Because the oscillator is stopped in STOP mode, a restart delay may be imposed to allow oscillator stabilization upon leaving STOP. If the internal oscillator is being used, this delay is required; however, if a stable external oscillator is being used, the DLY control bit can be used to bypass this start-up delay. The DLY control bit is set by reset and can be optionally cleared during initialization. If the DLY equal to zero option is used to avoid start-up delay on recovery from STOP, then reset should not be used as the means of recovering from STOP, as this causes DLY to be set again by reset, imposing the restart delay. This same delay also applies to power-on-reset, regardless of the state of the DLY control bit, but does not apply to a reset while the clocks are running. See Section 4.3.2.4.

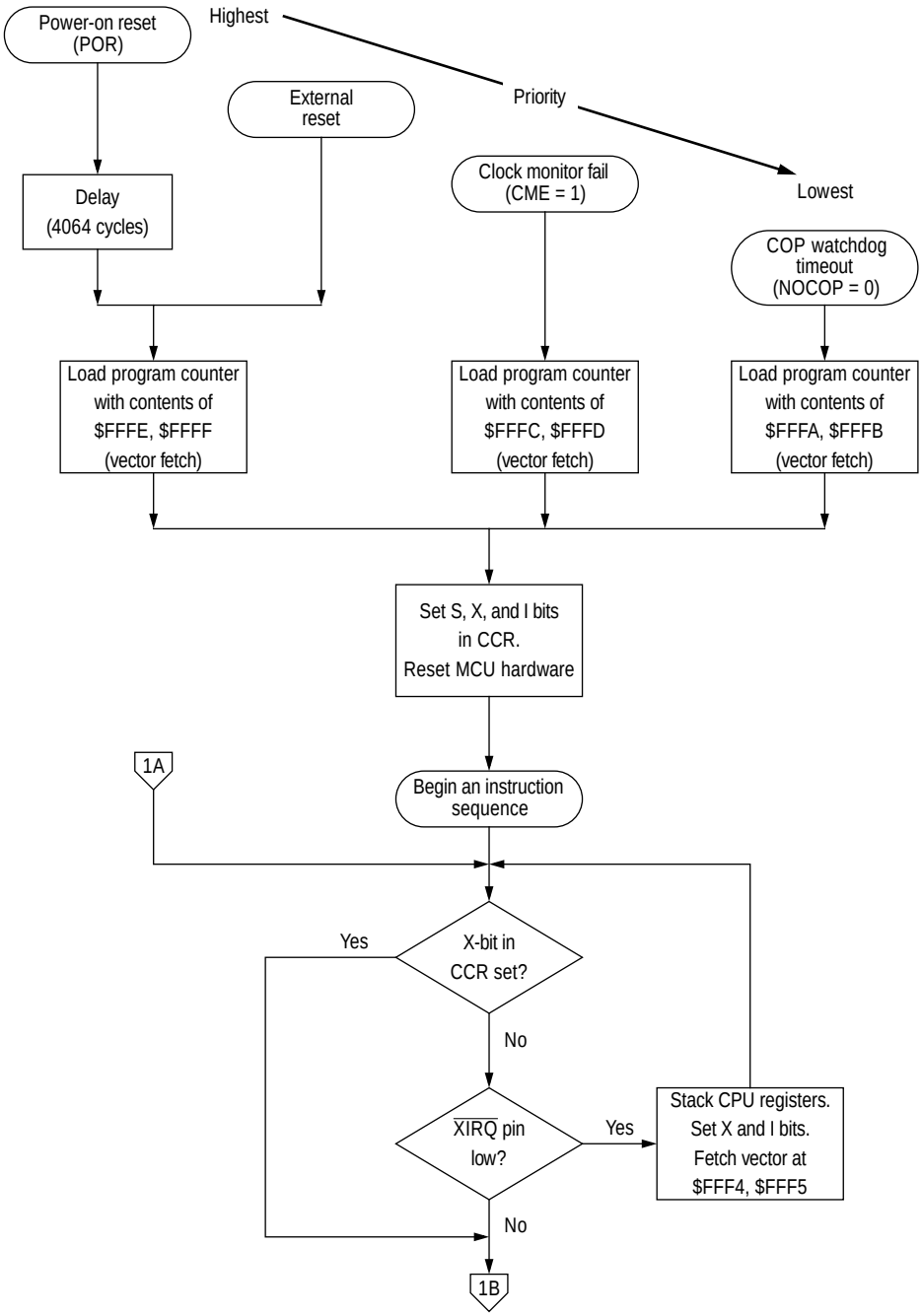


Figure 5-1 Processing flow out of reset (1 of 2)

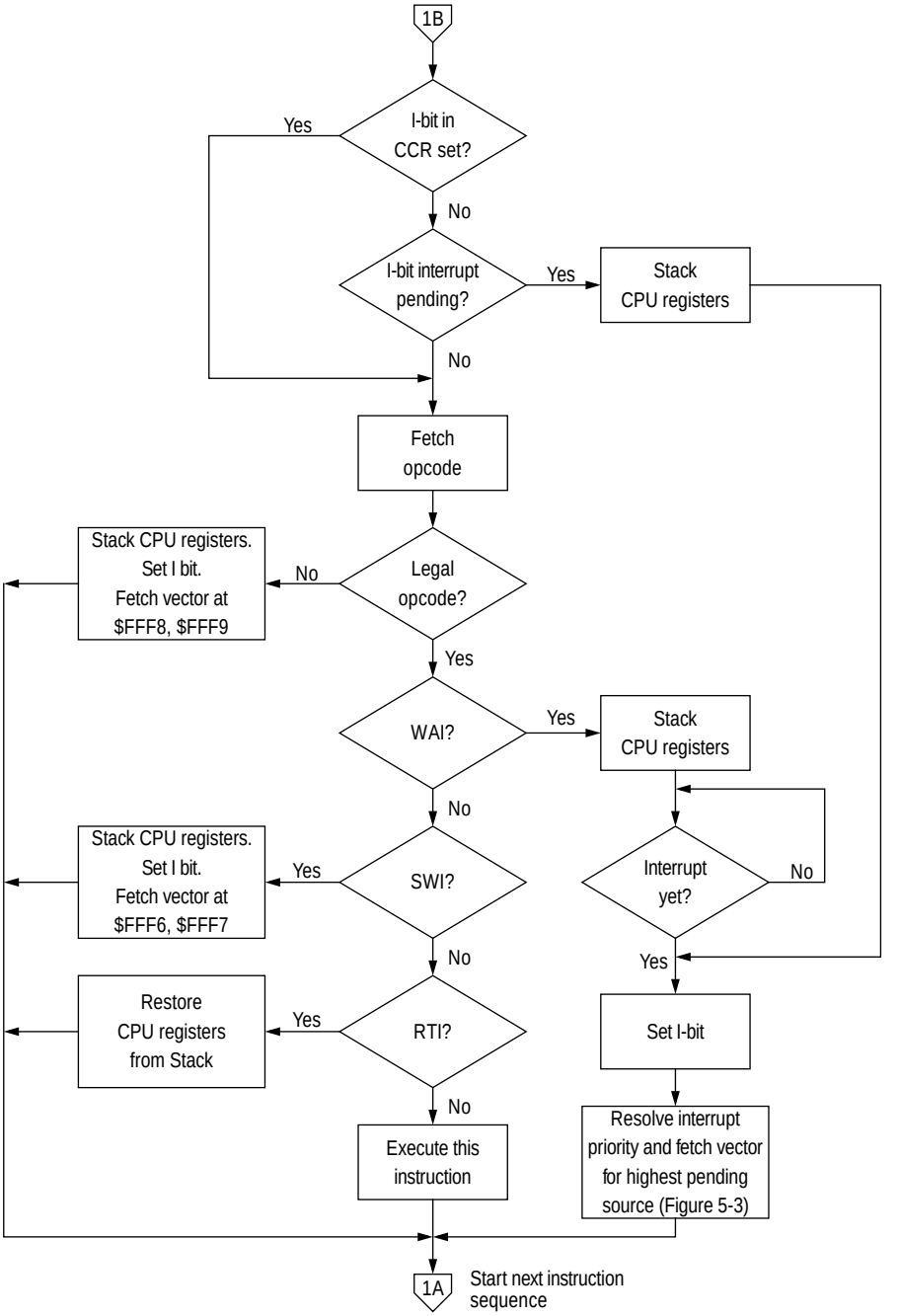


Figure 5-2 Processing flow out of reset (2 of 2)

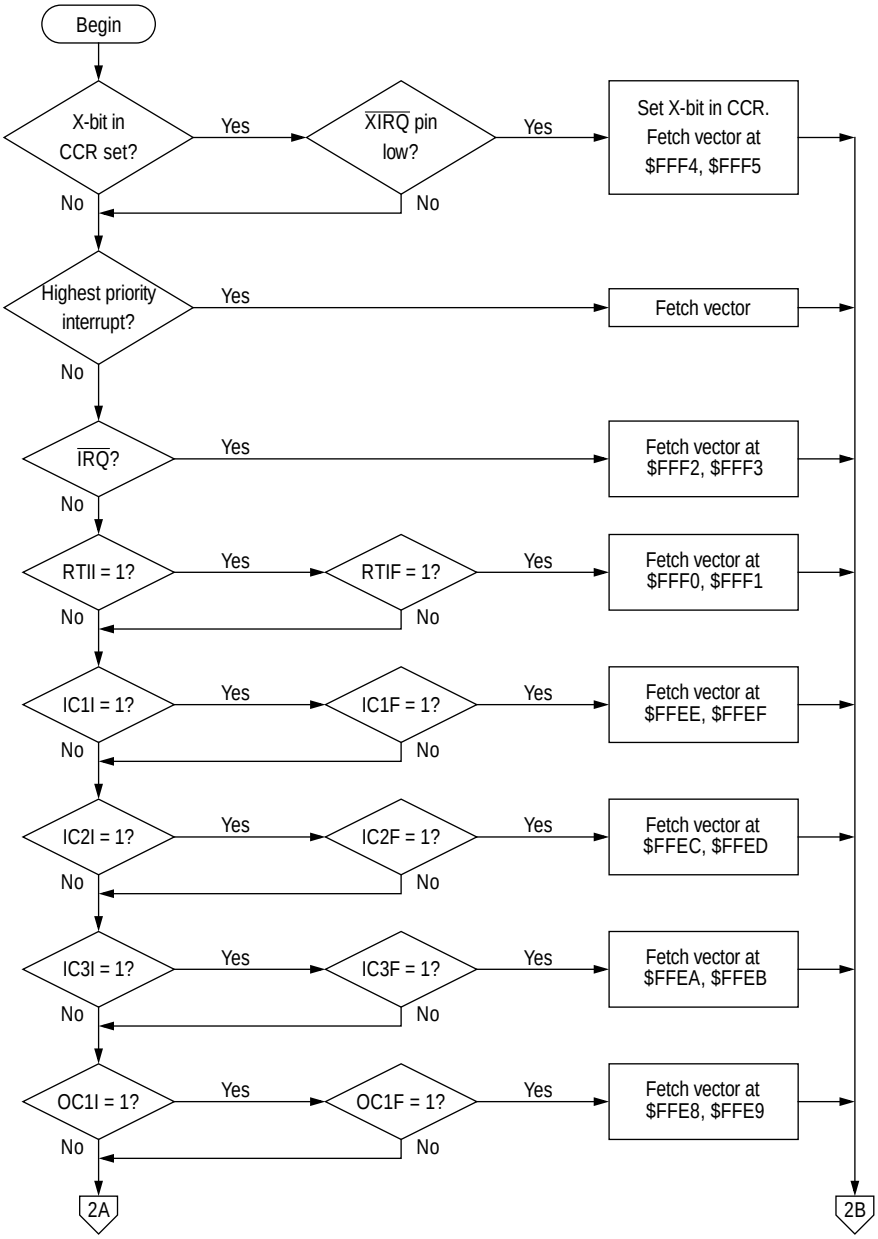
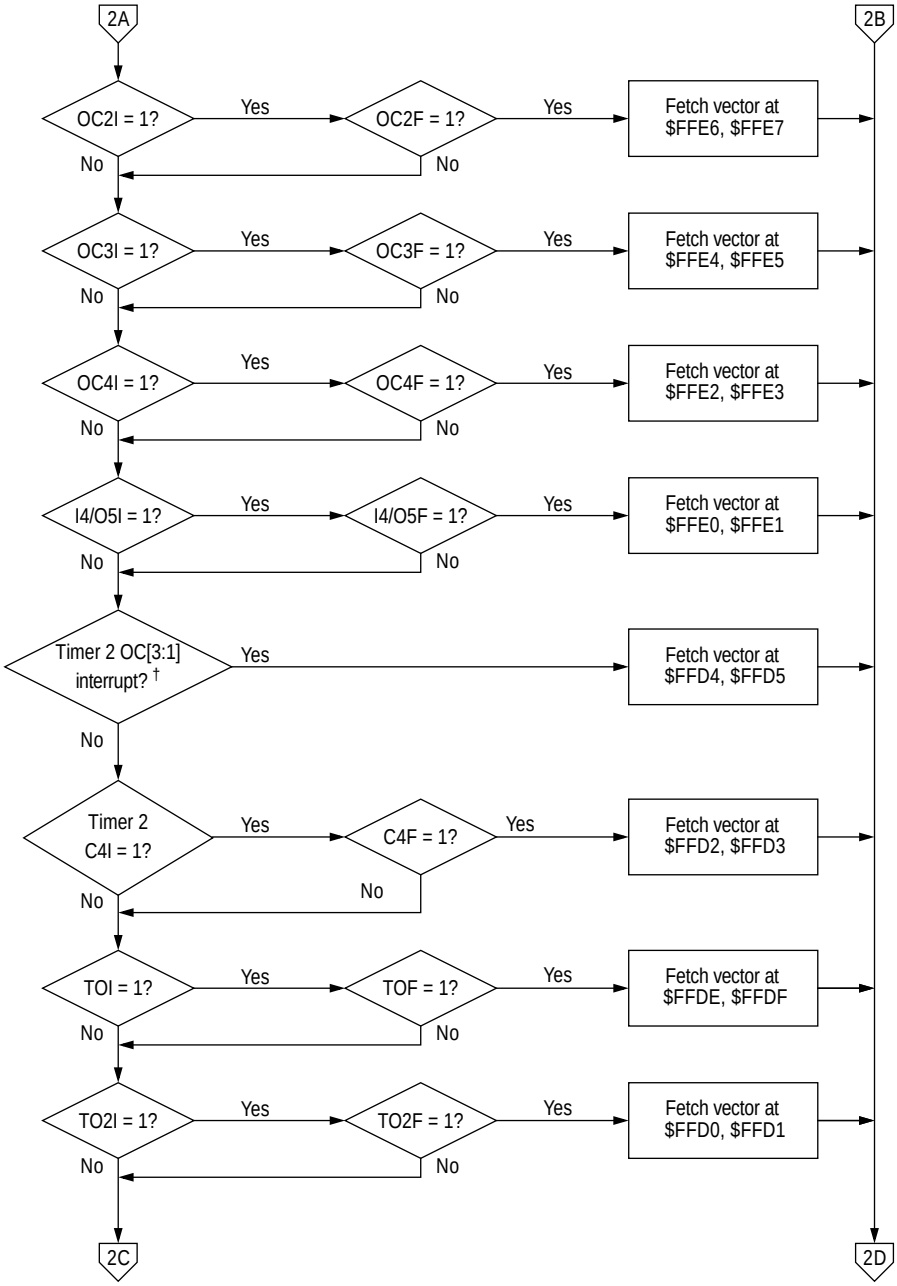
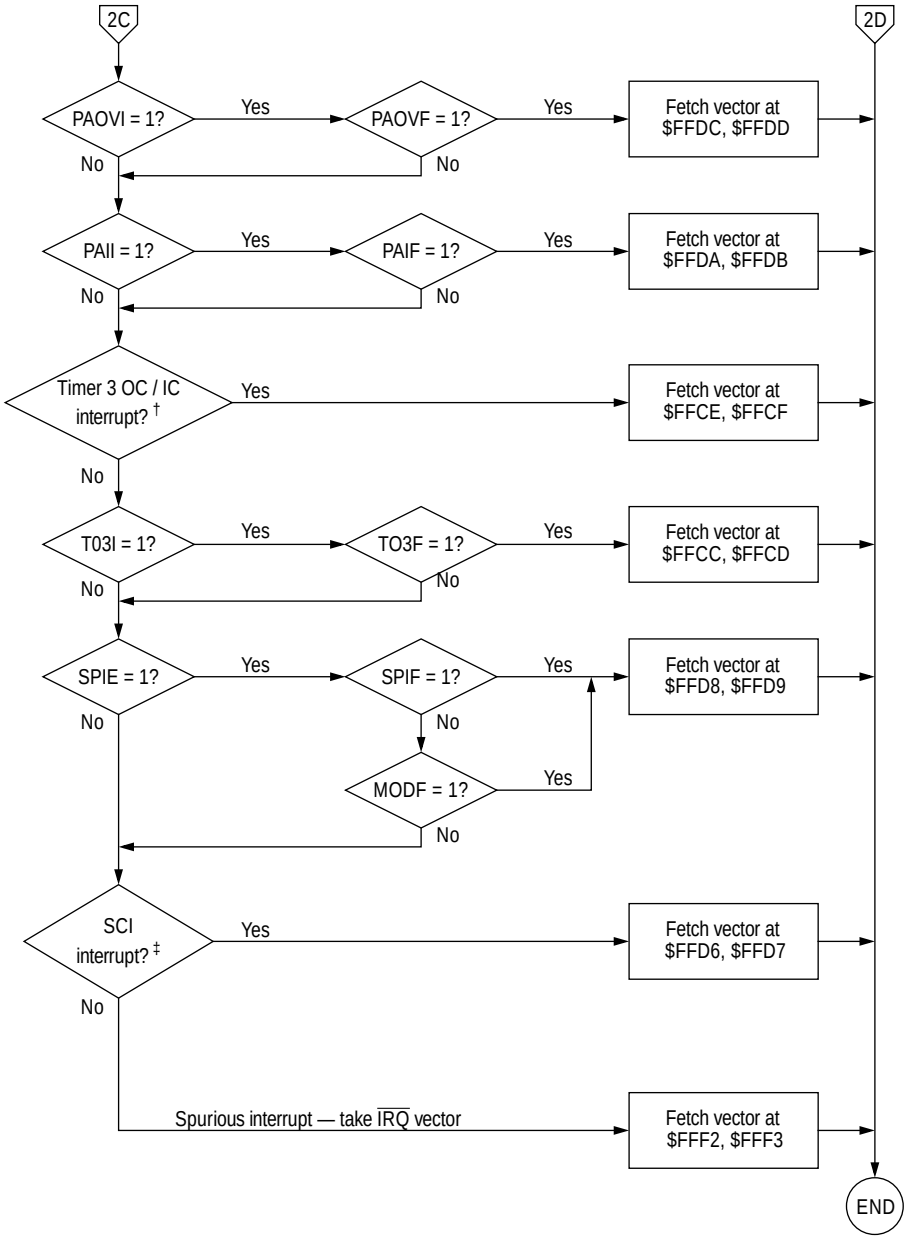


Figure 5-3 Interrupt priority resolution (1 of 3)



† Flag polling is required to determine the source of this interrupt (refer to Section 9).

Figure 5-4 Interrupt priority resolution (2 of 3)



† Flag polling is required to determine the source of this interrupt (refer to Section 9).

‡ Refer to Figure 5-6 for further details on SCI interrupts.

Figure 5-5 Interrupt priority resolution (3 of 3)

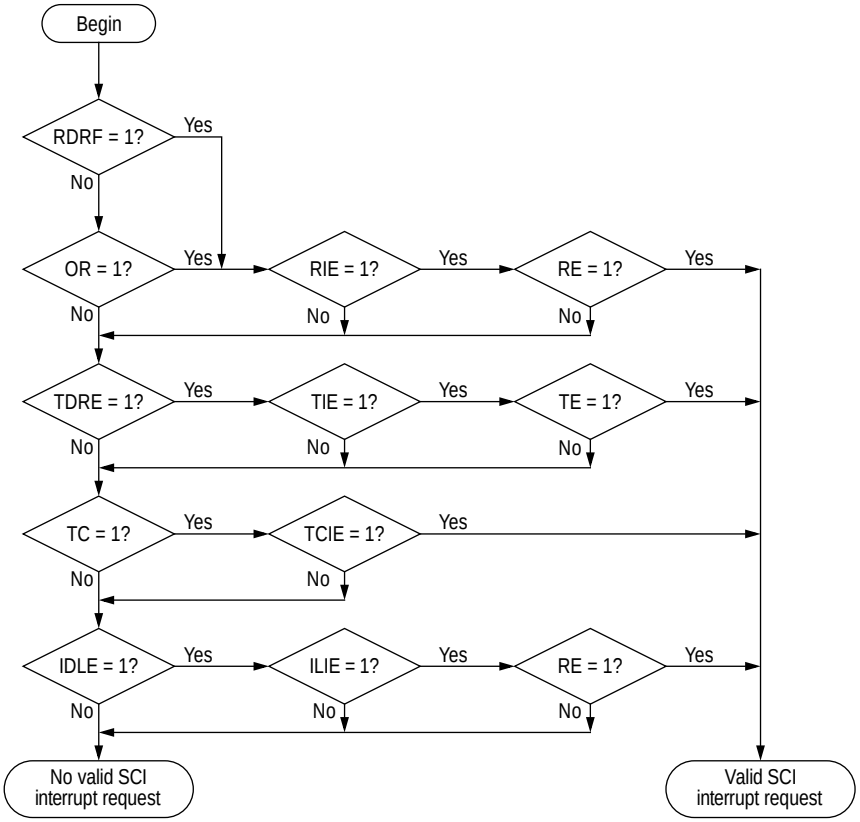


Figure 5-6 Interrupt source resolution within the SCI subsystem

6
PARALLEL INPUT/OUTPUT

The MC68HC11KW1 has up to 70 input/output lines and 10 input-only lines, depending on the operating mode. To enhance the I/O functions, the data bus of this microcontroller is non-multiplexed. The following table is a summary of the configuration and features of each port.

Table 6-1 Port configuration

Port	Input pins	Output pins	Bidirectional pins	Alternative functions
A	—	—	8	Timer 1
B	—	—	8	High order address
C	—	—	8	Data bus
D	—	—	8	SCI and SPI
E	8	—	–	A/D converter
F	—	—	8	Low order address
G	2	—	6	Memory expansion and A/D converter
H	—	—	8	Chip selects and PWM
J	—	—	8	Timer 2
K	—	—	8	Timer 3

Note: Do not confuse pin function with the electrical state of that pin at reset. All general-purpose I/O pins that are configured as inputs at reset are in a high-impedance state and the contents of the port data registers are undefined; in port descriptions, a ‘u’ indicates this condition. The pin function is mode dependent.

6.1 Port A

Port A is an 8-bit bidirectional port, with both data and data direction registers. In addition to their I/O capability, port A pins are shared with Timer 1 functions, as shown in the following table.

Pin	Alternative function
PA0	IC3
PA1	IC2
PA2	IC1
PA3	OC5 and/or OC1, or IC4
PA4	OC4 and/or OC1
PA5	OC3 and/or OC1
PA6	OC2 and/or OC1
PA7	PAI and/or OC1

See Section 9 for more information.

On reset the pins are configured as general purpose high-impedance inputs.

6.1.1 PORTA — Port A data register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Port A data (PORTA)	$0000	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0	undefined

This is a read/write register and is not affected by reset. The bits may be read and written at any time, but, when a pin is allocated to its alternate function, a write to the corresponding register bit has no effect on the pin state.

6.1.2 DDRA — Data direction register for port A

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Data direction A (DDRA)	$0001	DDA7	DDA6	DDA5	DDA4	DDA3	DDA2	DDA1	DDA0	0000 0000

DDA[7:0] — Data direction for port A

1 (set) – The corresponding pin is configured as an output.

0 (clear) – The corresponding pin is configured as an input.

6.2 Port B

Port B is an 8-bit bidirectional port, with both data and data direction registers. In addition to their I/O capability, port B pins are used as the non-multiplexed high order address pins, as shown in the following table.

Pin	Alternative function
PB0	ADDR8
PB1	ADDR9
PB2	ADDR10
PB3	ADDR11
PB4	ADDR12
PB5	ADDR13
PB6	ADDR14
PB7	ADDR15

In expanded or test mode, the pins become the high order address lines and port B is not included in the memory map.

The state of the pins on reset is mode dependent. In single chip or bootstrap mode, port B pins are high-impedance inputs with selectable internal pull-up resistors (see Section 6.11). In expanded or test mode, port B pins are high order address outputs and PORTB/DDRB are not in the memory map.

6.2.1 PORTB — Port B data register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Port B data (PORTB)	$0004	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0	undefined

These bits may be read and written at any time and are not affected by reset.

6.2.2 DDRB — Data direction register for port B

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Data direction B (DDRB)	$0002	DDB7	DDB6	DDB5	DDB4	DDB3	DDB2	DDB1	DDB0	0000 0000

DDB[7:0] — Data direction for port B

1 (set) – The corresponding pin is configured as an output.

0 (clear) – The corresponding pin is configured as an input.

6.3 Port C

Port C is an 8-bit bidirectional port, with both data and data direction registers. In addition to their I/O capability, port C pins are used as the non-multiplexed data bus pins, as shown in the following table.

Pin	Alternative function
PC0	DATA0
PC1	DATA1
PC2	DATA2
PC3	DATA3
PC4	DATA4
PC5	DATA5
PC6	DATA6
PC7	DATA7

In expanded or test mode, the pins become the data bus and port C is not included in the memory map.

The state of the pins on reset is mode dependent. In single chip or bootstrap mode, port C pins are high-impedance inputs. In expanded or test modes, port C pins are the data bus I/O and PORTC/DDRC are not in the memory map.

6.3.1 PORTC — Port C data register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Port C data (PORTC)	$0006	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0	undefined

The bits may be read and written at any time and are not affected by reset.

6.3.2 DDRC — Data direction register for port C

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Data direction C (DDRC)	$0007	DDC7	DDC6	DDC5	DDC4	DDC3	DDC2	DDC1	DDC0	0000 0000

DDC[7:0] — Data direction for port C

1 (set) – The corresponding pin is configured as an output.

0 (clear) – The corresponding pin is configured as an input.

6.4 Port D

Port D is an 8-bit bidirectional port, with both data and data direction registers. In addition to their I/O capability, six of port D's pins are shared with SCI and SPI functions, as shown in the following table.

Pin	Alternative function	
PD0	RXD	See Section 7 for more information.
PD1	TXD	
PD2	MISO	See Section 8 for more information.
PD3	MOSI	
PD4	SCK	
PD5	$\overline{SS}$	

On reset the pins are configured as general purpose high-impedance inputs.

6.4.1 PORTD — Port D data register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Port D data (PORTD)	$0008	PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0	undefined

This is a read/write register and is not affected by reset. The bits may be read and written at any time, but, when a pin is allocated to an alternate function, a write to the corresponding register bit has no effect on the pin state.

6.4.2 DDRD — Data direction register for port D

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Data direction D (DDRD)	$0009	DDD7	DDD6	DDD5	DDD4	DDD3	DDD2	DDD1	DDD0	0000 0000

DDD[7:0] — Data direction for port D

1 (set) – The corresponding pin is configured as an output.

0 (clear) – The corresponding pin is configured as an input.

6.5 Port E

Port E is an input-only port. In addition to their input capability, port E pins are shared with A/D functions, as shown in the following table.

Pin	Alternative function
PE0	AN2
PE1	AN3
PE2	AN4
PE3	AN5
PE4	AN6
PE5	AN7
PE6	AN8
PE7	AN9

See Section 10 for more information.

On reset, the pins are configured as general purpose high-impedance inputs.

6.5.1 PORTE — Port E data register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Port E data (PORTE)	$000A	PE7	PE6	PE5	PE4	PE3	PE2	PE1	PE0	undefined

This is a read-only register and is not affected by reset. The bits may be read at any time.

Note: As port E shares pins with the A/D converter, a read of this register may affect any conversion currently in progress, if it coincides with the sample portion of the conversion cycle. Hence, normally port E should not be read during the sample portion of any conversion.

6.6 Port F

Port F is an 8-bit bidirectional port, with both data and data direction registers. In addition to their I/O capability, port F pins are used as the non-multiplexed low order address pins, as shown in the following table.

Pin	Alternative function
PF0	ADDR0
PF1	ADDR1
PF2	ADDR2
PF3	ADDR3
PF4	ADDR4
PF5	ADDR5
PF6	ADDR6
PF7	ADDR7

In expanded or test mode, the pins become the low order address and port F is not included in the memory map.

The state of the pins on reset is mode dependent. In single chip or bootstrap mode, port F pins are high-impedance inputs with selectable internal pull-up resistors (see Section 6.11). In expanded or test modes, port F pins are low order address outputs and PORTF/DDRF are not in the memory map.

6.6.1 PORTF — Port F data register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Port F data (PORTF)	$0005	PF7	PF6	PF5	PF4	PF3	PF2	PF1	PF0	undefined

The bits may be read and written at any time and are not affected by reset.

6.6.2 DDRF — Data direction register for port F

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Data direction F (DDRF)	$0003	DDF7	DDF6	DDF5	DDF4	DDF3	DDF2	DDF1	DDF0	0000 0000

DDF[7:0] — Data direction for port F

1 (set) – The corresponding pin is configured as an output.

0 (clear) – The corresponding pin is configured as an input.

6.7 Port G

Port G is an 8-bit port, with both data and data direction registers. Pins [7, 6] are input-only, and may be used as general purpose inputs, or as inputs to the A/D converter. Pins [5:0] are fully bidirectional, and, in addition to their I/O capability, are shared with memory expansion functions, as shown in the following table. The functions of pins [5:0] are controlled by bits in the port G assignment register, PGAR.

Pin	Alternative function	
PG0	XA13	See Section 4 for more information.
PG1	XA14	
PG2	XA15	
PG3	XA16	
PG4	XA17	
PG5	XA18	
PG6	AN0	See Section 10 for more information.
PG7	AN1	

Note: The input timing characteristics of pins PG[7, 6] are different from the other port G pins. Refer to Section A.5.1.

6.7.1 PORTG — Port G data register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Port G data (PORTG)	$007E	PG7	PG6	PG5	PG4	PG3	PG2	PG1	PG0	undefined

This is a read/write register and is not affected by reset. The bits may be read and written at any time, but, when a pin is allocated to its alternate function, a write to the corresponding register bit has no effect on the pin state. Port G pins [5:0] have internal, software-selectable pull-up resistors which are controlled by the PPAR register. See Section 6.11.1.

Note: As port G shares two pins with the A/D converter, a read of this register may affect any conversion currently in progress, if it coincides with the sample portion of the conversion cycle. Hence, normally port G should not be read during the sample portion of any conversion.

6.7.2 DDRG — Data direction register for port G

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Data direction G (DDRG)	$007F	0	0	DDG5	DDG4	DDG3	DDG2	DDG1	DDG0	0000 0000

Bits [7, 6] — Not implemented; always read zero

DDG[5:0] — Data direction for port G

1 (set) – The corresponding pin is configured as an output.

0 (clear) – The corresponding pin is configured as an input.

6.7.3 PGAR — Port G assignment register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Port G assignment (PGAR)	$002D	0	0	PGAR5	PAGR4	PGAR3	PGAR2	PGAR1	PGAR0	0000 0000

PGAR selects which port G pins are used for I/O or memory expansion address lines, defining which extended address lines are used. Selecting an address on one of these pins causes a port G pin to be lost. For this reason, select only those address lines that are needed by the expansion logic. This allows unused lines to serve as general-purpose I/O. For more information, refer to Section 4.4.

Bits [7:6] — Not implemented; always read zero

PGAR[5:0] — Port G pin assignment

1 (set) – Corresponding port G pin is expansion address line (XA[18:13]).

0 (clear) – Corresponding port G pin is general-purpose I/O.

6.8 Port H

Port H is an 8-bit, bidirectional port, with both data and data direction registers. In addition to their I/O capability, port H pins are shared with chip select and PWM functions, as shown in the following table.

Pin	Alternative function	
PH7	CSPROG	See Section 4 for more information.
PH6	CSGP2	
PH5	CSGP1	
PH4	CSIO	
PH3	PWM4	See Section 9 for more information.
PH2	PWM3	
PH1	PWM2	
PH0	PWM1	

6.8.1 PORTH — Port H data register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Port H data (PORTH)	$007C	PH7	PH6	PH5	PH4	PH3	PH2	PH1	PH0	undefined

This is a read/write register and is not affected by reset.

The bits may be read and written at any time, but, when a pin is allocated to its alternate function, a write to the corresponding register bit has no effect on the pin state. Port H has internal, software selectable pull-up resistors which are controlled by the PPAR register. See Section 6.11.1.

6.8.2 DDRH — Data direction register for port H

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Data direction H (DDRH)	$007D	DDH7	DDH6	DDH5	DDH4	DDH3	DDH2	DDH1	DDH0	0000 0000

DDH[7:0] — Data direction for port H

1 (set) – The corresponding pin is configured as an output.

0 (clear) – The corresponding pin is configured as an input.

6.9 Port J

Port J is an 8-bit, bidirectional port, with both data and data direction registers. In addition to their I/O capabilities, five of port J's pins are shared with Timer 2 functions, as shown in the following table.

Pin	Alternative function
PJ7	C4
PJ6	OC3
PJ5	OC2
PJ4	OC1
PJ3	ECIN

See Section 9 for more information.

6.9.1 PORTJ — Port J data register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Port J data (PORTJ)	$008E	PJ7	PJ6	PJ5	PJ4	PJ3	PJ2	PJ1	PJ0	undefined

This is a read/write register and is not affected by reset. These bits may be read and written at any time, but, when a pin is allocated to its alternate function, a write to the corresponding register bit has no effect on the pin state.

6.9.2 DDRJ — Data direction register for port J

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Data direction J (DDRJ)	$008F	DDJ7	DDJ6	DDJ5	DDJ4	DDJ3	DDJ2	DDJ1	DDJ0	0000 0000

DDJ[7:0] — Data direction for port J

1 (set) – The corresponding pin is configured as an output.

0 (clear) – The corresponding pin is configured as an input.

6.10 Port K

Port K is an 8-bit bidirectional port, with both data and data direction registers. In addition to their I/O capabilities, five of port K's pins are shared with Timer 3 functions, as shown in the following table.

Pin	Alternative function
PK7	C4
PK6	OC3
PK5	OC2
PK4	OC1
PK3	ECIN

See Section 9 for more information.

6.10.1 PORTK — Port K data register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Port K data (PORTK)	$009E	PK7	PK6	PK5	PK4	PK3	PK2	PK1	PK0	undefined

This is a read/write register and is not affected by reset.

The bits may be read and written at any time, but, when a pin is allocated to its alternate function, a write to the corresponding register bit has no effect on the pin state.

6.10.2 DDRK — Data direction register for port K

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Data direction K (DDRK)	$009F	DDK7	DDK6	DDK5	DDK4	DDK3	DDK2	DDK1	DDK0	0000 0000

DDK[7:0] — Data direction for port K

1 (set) – The corresponding pin is configured as an output.

0 (clear) – The corresponding pin is configured as an input.

6.11 Internal pull-up resistors

Four of the ports (B, F, G and H) have internal, software selectable pull-up resistors under control of the port pull-up assignment register (PPAR).

6.11.1 PPAR — Port pull-up assignment register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Port pull-up assignment (PPAR)	$002C	0	0	0	0	HPPUE	GPPUE	FPPUE	BPPUE	0000 1111

Bits [7:4] — Not implemented; always read zero.

xPPUE — Port x pin pull-up enable

These bits control the on-chip pull-up devices connected to all the pins on I/O ports B, F, H and PG[5:0]. They are collectively enabled or disabled via the PAREN bit in the CONFIG register (see Section 6.12.2).

1 (set) – Port x pin on-chip pull-up devices enabled.

0 (clear) – Port x pin on-chip pull-up devices disabled.

Note: FPPUE and BPPUE have no effect in expanded mode since ports F and B are then dedicated address bus outputs.

Note: HPPUE and GPPUE are set on reset, to insure that all expanded memory address signals and chip select signals will be pulled to a logic high level (since the pins are configured for general I/O and set as high impedance inputs).

Note: The pull-up resistors are disabled when the PAREN bit in the CONFIG register is equal to ‘0’. The approximate value of these resistors is 14–17Kohms.

6.12 System configuration

One bit in each of the following registers is directly concerned with the configuration of the I/O ports. For full details on the other bits in the registers, refer to the appropriate section.

6.12.1 OPT2 — System configuration options register 2

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
System config. options 2 (OPT2)	$0038	LIRDV	CWOM	0	IRVNE	LSBF	SPR2	XDV1	XDV0	000x 0000

LIRDV — $\overline{\text{LIR}}$ driven (refer to Section 4)

1 (set) – Enable $\overline{\text{LIR}}$ drive high pulse.

0 (clear) – $\overline{\text{LIR}}$ not driven high on MODA/$\overline{\text{LIR}}$ pin.

CWOM — Port C wired-OR mode

1 (set) – Port C outputs are open-drain.

0 (clear) – Port C operates normally.

Bit 5 — Not implemented; always reads zero.

IRVNE — Internal read visibility/not E (refer to Section 4)

1 (set) – Data from internal reads is driven out of the external data bus.

0 (clear) – No visibility of internal reads on external bus.

In **single chip mode** this bit determines whether the E clock drives out from the chip.

1 (set) – E pin is driven low.

0 (clear) – E clock is driven out from the chip.

LSBF — LSB first enable (refer to Section 8)

1 (set) – SPI data is transferred LSB first.

0 (clear) – SPI data is transferred MSB first.

SPR2 — SPI clock rate select (refer to Section 8)

XDV[1, 0] — XOUT clock divide select (refer to Section 4)

These two bits control the frequency of the XCLK signal, which is output on the XOUT pin if enabled by the CLKX bit in CONFIG.

6.12.2 CONFIG — System configuration register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Configuration control (CONFIG)	$003F	1	1	CLKX	PAREN	NOSEC	NOCOP	1	EEON	11xx xx1x

Bits [7, 6, 1] — Not implemented; always reads as one.

CLKX — X clock enable (refer to Section 4)

1 (set) – XCLK signal is driven out on the XOUT pin.

0 (clear) – XOUT pin is disabled.

PAREN — Pull-up assignment register enable

1 (set) – Pull-ups can be enabled using PPAR register.

0 (clear) – All pull-ups disabled.

NOSEC — EEPROM security disabled (refer to Section 4)

1 (set) – Disable security.

0 (clear) – Enable security.

NOCOP — COP system disable (refer to Section 5)

1 (set) – COP system disabled.

0 (clear) – COP system enabled (forces reset on timeout).

EEON — EEPROM enable (refer to Section 4)

1 (set) – EEPROM is present in the memory map.

0 (clear) – EEPROM is disabled from the memory map.

THIS PAGE LEFT BLANK INTENTIONALLY

7
SERIAL COMMUNICATIONS INTERFACE

The serial communications interface (SCI) is a universal asynchronous receiver transmitter (UART). It has a non-return to zero (NRZ) format (one start, eight or nine data, and one stop bit) that is compatible with standard RS-232 systems.

The SCI shares I/O with two of port D's pins:

Pin	Alternative function
PD0	RXD
PD1	TXD

The SCI transmit and receive functions are enabled by TE and RE respectively, in SCCR2.

The SCI features enabled on this MCU include: 13-bit modulus prescaler, idle line detect, receiver-active flag, transmitter and receiver hardware parity. A block diagram of the enhanced baud rate generator is shown in Figure 7-1. See Table 7-1 for example baud rate control values.

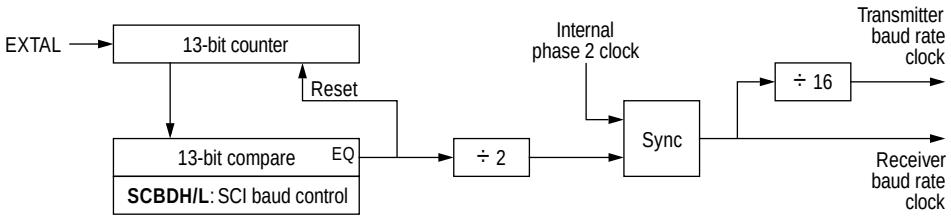


Figure 7-1 SCI baud rate generator circuit diagram

7.1 Data format

The serial data format requires the following conditions:

- An idle-line condition before transmission or reception of a message.
- A start bit, logic zero, transmitted or received, that indicates the start of each character.
- Data that is transmitted and received least significant bit (LSB) first.
- A stop bit, logic one, used to indicate the end of a frame. (A frame consists of a start bit, a character of eight or nine data bits, and a stop bit.)
- A break (defined as the transmission or reception of a logic zero for some multiple number of frames).

Selection of the word length is controlled by the M bit of SCCR1.

7.2 Transmit operation

The SCI transmitter includes a parallel data register (SCDRH/SCDRL) and a serial shift register. The contents of the shift register can only be written through the parallel data register. This double buffered operation allows a character to be shifted out serially while another character is waiting in the parallel data register to be transferred into the shift register. The output of the shift register is applied to TXD as long as transmission is in progress or the transmit enable (TE) bit of serial communication control register 2 (SCCR2) is set. The block diagram, Figure 7-2, shows the transmit serial shift register and the buffer logic at the top of the figure.

7.3 Receive operation

During receive operations, the transmit sequence is reversed. The serial shift register receives data and transfers it to the parallel receive data registers (SCDRH/SCDRL) as a complete word. This double buffered operation allows a character to be shifted in serially while another character is still in the serial data registers. An advanced data recovery scheme distinguishes valid data from noise in the serial data stream. The data input is selectively sampled to detect receive data, and majority sampling logic determines the value and integrity of each bit.

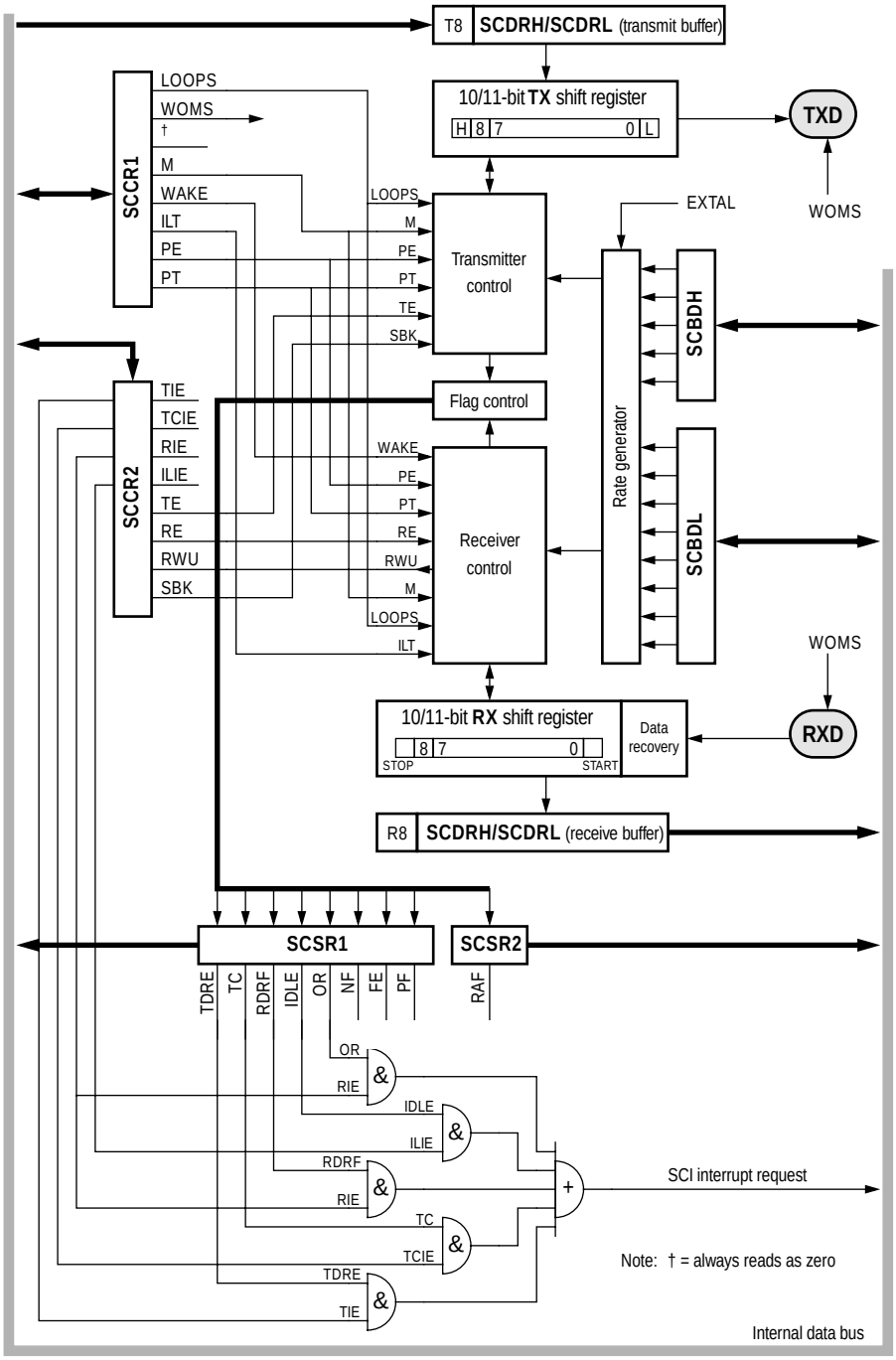


Figure 7-2 SCI block diagram

7.4 Wake-up feature

The wake-up feature reduces SCI service overhead in multiple receiver systems. Software for each receiver evaluates the first character or frame of each message. All receivers are placed in wake-up mode by writing a one to the RWU bit in the SCCR2 register. When RWU is set, the receiver-related status flags (RDRF, IDLE, OR, NF, FE, and PF) are inhibited (cannot be set). Although RWU can be cleared by a software write to SCCR2, to do so would be unusual. Normally RWU is set by software and is cleared automatically with hardware. Whenever a new message begins, logic alerts the dormant receivers to wake up and evaluate the initial character of the new message.

Two methods of wake-up are available: idle-line wake-up and address mark wake-up. During idle-line wake-up, a dormant receiver activates as soon as the RXD line becomes idle. In the address mark wake-up, logic one in the most significant bit (MSB) of a character activates all sleeping receivers. To use either receiver wake-up method, establish a software addressing scheme to allow the transmitting devices to direct messages to individual receivers or to groups of receivers. This addressing scheme can take any form as long as all transmitting and receiving devices are programmed to understand the same scheme.

7.4.1 Idle-line wake-up

Clearing the WAKE bit in SCCR1 register enables idle-line wake-up mode. In idle-line wake-up mode, all receivers are active (RWU bit in SCCR2 = 0) when each message begins. The first frames of each message are addressing frames. Each receiver in the system evaluates the addressing frames of a message to determine if the message is intended for that receiver. When a receiver finds that the message is not intended for it, it sets the RWU bit. Once set, the RWU control bit disables all but the necessary receivers for the remainder of the message, thus reducing software overhead for the remainder of that message. As soon as an idle line is detected by receiver logic, hardware automatically clears the RWU bit so that the first frames of the next message can be evaluated by all receivers in the system. This type of receiver wake-up requires a minimum of one idle frame time between messages, and no idle time between frames within a message.

7.4.2 Address-mark wake-up

Setting the WAKE bit in SCCR1 register enables address-mark wake-up mode. The address-mark wake-up method uses the MSB of each frame to differentiate between address information (MSB = 1) and actual message data (MSB = 0). All frames consist of seven information bits (eight bits if M bit in SCCR1 = 1) and an MSB which, when set to one, indicates an address frame. The first frames of each message are addressing frames. Receiver logic evaluates these marked frames to determine the receivers for which that message is intended. When a receiver finds that the message is not intended for it, it sets the RWU bit. Once set, the RWU control bit disables all but the necessary receivers for the remainder of the message, thus reducing software overhead

for the remainder of that message. When the next message begins, its first frame will have the MSB set which will automatically clear the RWU bit and indicate that this is an addressing frame. This frame is always the first frame received after wake-up because the RWU bit is cleared before the stop bit for the first frame is received. This method of wake-up allows messages to include idle times, however, there is a loss in efficiency due to the extra bit time required for the address bit in each frame.

7.5 SCI error detection

Four error conditions can occur during SCI operation. These error conditions are: serial data register overrun, received bit noise, framing, and parity error. Four bits (OR, NF, FE, and PF) in serial communications status register 1 (SCSR1) indicate if one of these error conditions exists.

The overrun error (OR) bit is set when the next byte is ready to be transferred from the receive shift register to the serial data registers (SCDRH/SCDRL) and the registers are already full (RDRF bit is set). When an overrun error occurs, the data that caused the overrun is lost and the data that was already in serial data registers is not disturbed. The OR is cleared when the SCSR is read (with OR set), followed by a read of the SCI data registers.

The noise flag (NF) bit is set if there is noise on any of the received bits, including the start and stop bits. The NF bit is not set until the RDRF flag is set. The NF bit is cleared when the SCSR is read (with FE equal to one) followed by a read of the SCI data registers.

When no stop bit is detected in the received data character, the framing error (FE) bit is set. FE is set at the same time as the RDRF. If the byte received causes both framing and overrun errors, the processor only recognizes the overrun error. The framing error flag inhibits further transfer of data into the SCI data registers until it is cleared. The FE bit is cleared when the SCSR is read (with FE equal to one) followed by a read of the SCI data registers.

The parity error flag (PF) is set if received data has incorrect parity. The flag is cleared by a read of SCSR1 with PE set, followed by a read of SCDR.

7.6 SCI registers

There are eight addressable registers in the SCI. SCBDH, SCBDL, SCCR1 and SCCR2 are control registers. The contents of these registers control functions and indicate conditions within the SCI. The status registers SCSR1 and SCSR2 contain bits that indicate certain conditions within the SCI. SCDRH and SCDRL are SCI data registers. These double buffered registers are used for the transmission and reception of data, and are used to form the 9-bit data word for the SCI. If the SCI is being used with 7 or 8-bit data, only SCDRL needs to be accessed. Note that if 9-bit data 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.

7.6.1 SCBDH, SCBDL — SCI baud rate control registers

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
SCI baud rate high (SCBDH)	$0070	BTST	BSPL	SYNC	SBR12	SBR11	SBR10	SBR9	SBR8	0000 0000
SCI baud rate low (SCBDL)	$0071	SBR7	SBR6	SBR5	SBR4	SBR3	SBR2	SBR1	SBR0	0000 0100

The contents of this register determine the baud rate of the SCI.

BTST — Baud register test (Test mode only)

BSPL — Baud rate counter split (Test mode only)

SYNC — Baud rate counter reset and sync (Test mode only)

SBR[12:0] — SCI baud rate selects

Use the following formula to calculate SCI baud rate. Refer to the table of baud rate control values for example rates:

$$\text{SCI baud rate} = \frac{\text{EXTAL}}{16 \times (2\text{BR})}$$

where the baud rate control value (BR) is the contents of SCBDH/L (BR = 1, 2, 3,... 8191).

For example, to obtain a baud rate of 1200 with an EXTAL frequency of 16 MHz, the baud register (SCBDH/L) should contain $01A0 (see Table 7-1).

The clock rate generator is disabled if BR = 0, or if neither the receiver nor transmitter is enabled (both RE and TE in SCCR2 are cleared).

Writes to the baud rate registers will only be successful if the last (or only) byte written is SCBDL. The use of an STD instruction is recommended as it guarantees that the bytes are written in the correct order.

Table 7-1 Example SCI baud rate control values

Target baud rate	EXTAL frequency: 16 MHz	
	Dec value	Hex value
110	4545	$11C1
150	3333	$0D05
300	1666	$0682
600	833	$0341
1200	416	$01A0
2400	208	$00D0
4800	104	$0068
9600	52	$0034
19200	26	$001A
38400	13	$000D

7.6.2 SCCR1 — SCI control register 1

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
SCI control 1 (SCCR1)	$0072	LOOPS	WOMS	0	M	WAKE	ILT	PE	PT	0000 0000

The SCCR1 register provides the control bits that determine word length and select the method used for the wake-up feature.

LOOPS — SCI loop mode enable

1 (set) – SCI transmit and receive are disconnected from TXD and RXD pins, and transmitter output is fed back into the receiver input.

0 (clear) – SCI transmit and receive operate normally.

Both the transmitter and receiver must be enabled to use the LOOP mode. When the LOOP mode is enabled, the TXD pin is driven high (idle line state) if the transmitter is enabled.

WOMS — Wired-OR mode for SCI pins (PD1, PD0)

1 (set) – TXD and RXD are open drains if operating as outputs.

0 (clear) – TXD and RXD operate normally.

Bit 5 — Not implemented; always reads zero

M — Mode (select character format)

1 (set) – Start bit, 9 data bits, 1 stop bit.

0 (clear) – Start bit, 8 data bits, 1 stop bit.

WAKE — Wake-up by address mark/idle

1 (set) – Wake-up by address mark (most significant data bit set).

0 (clear) – Wake-up by IDLE line recognition.

ILT — Idle line type

1 (set) – Long (SCI counts ones only after stop bit).

0 (clear) – Short (SCI counts consecutive ones after start bit).

This bit determines which of two types of idle line detection method is used by the SCI receiver. In short mode the stop bit and any bits that were ones before the stop bit will be considered as part of that string of ones, possibly resulting in erroneous or premature detection of an idle line condition. In long mode the SCI system does not begin counting ones until a stop bit is received.

PE — Parity enable

1 (set) – Parity enabled.

0 (clear) – Parity disabled.

PT — Parity type

1 (set) – Parity odd (an odd number of ones causes parity bit to be zero, an even number of ones causes parity bit to be one).

0 (clear) – Parity even (an even number of ones causes parity bit to be zero, an odd number of ones causes parity bit to be one).

7.6.3 SCCR2 — SCI control register 2

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
SCI control 2 (SCCR2)	$0073	TIE	TCIE	RIE	ILIE	TE	RE	RWU	SBK	0000 0000

The SCCR2 register provides the control bits that enable or disable individual SCI functions.

TIE — Transmit interrupt enable

1 (set) – SCI interrupt requested when TDRE status flag is set.

0 (clear) – TDRE interrupts disabled.

TCIE — Transmit complete interrupt enable

1 (set) – SCI interrupt requested when TC status flag is set.

0 (clear) – TC interrupts disabled.

RIE — Receiver interrupt enable

1 (set) – SCI interrupt requested when RDRF flag or the OR status flag is set.

0 (clear) – RDRF and OR interrupts disabled.

ILIE — Idle line interrupt enable

1 (set) – SCI interrupt requested when IDLE status flag is set.

0 (clear) – IDLE interrupts disabled.

TE — Transmitter enable

1 (set) – Transmitter enabled.

0 (clear) – Transmitter disabled.

RE — Receiver enable

1 (set) – Receiver enabled.

0 (clear) – Receiver disabled.

RWU — Receiver wake-up control

1 (set) – Wake-up enabled and receiver interrupts inhibited.

0 (clear) – Normal SCI receiver.

SBK — Send break

1 (set) – Break codes generated as long as SBK is set.

0 (clear) – Break generator off.

7.6.4 SCSR1 — SCI status register 1

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
SCI status 1 (SCSR1)	$0074	TDRE	TC	RDRF	IDLE	OR	NF	FE	PF	1100 0000

The bits in SCSR1 indicate certain conditions in the SCI hardware and are automatically cleared by special acknowledge sequences.

TDRE — Transmit data register empty flag

1 (set) – SCDR empty.

0 (clear) – SCDR busy.

This flag is set when SCDR is empty. Clear the TDRE flag by reading SCSR1 with TDRE set and then writing to SCDR.

TC — Transmit complete flag

1 (set) – Transmitter idle.

0 (clear) – Transmitter busy.

This flag is set when the transmitter is idle (no data, preamble, or break transmission in progress). Clear the TC flag by reading SCSR1 with TC set and then writing to SCDR.

RDRF — Receive data register full flag

1 (set) – SCDR full.

0 (clear) – SCDR empty.

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 SCDR. Clear the RDRF flag by reading SCSR1 with RDRF set and then reading SCDR.

IDLE — Idle line detected flag

1 (set) – RXD line is idle.

0 (clear) – RXD line is active.

This flag is set if the RXD line is idle. Once cleared, IDLE is not set again until the RXD line has been active and becomes idle again. The IDLE flag is inhibited when RWU = 1. Clear IDLE by reading SCSR1 with IDLE set and then reading SCDR.

OR — Overrun error flag

1 (set) – Overrun detected.

0 (clear) – No overrun.

OR is set if a new character is received before a previously received character is read from SCDR. Clear the OR flag by reading SCSR1 with OR set and then reading SCDR.

NF — Noise error flag

1 (set) – Noise detected.

0 (clear) – Unanimous decision.

NF is set if the majority sample logic detects anything other than a unanimous decision. Clear NF by reading SCSR1 with NF set and then reading SCDR.

FE — Framing error

1 (set) – Zero detected.

0 (clear) – Stop bit detected.

FE is set when a zero is detected where a stop bit was expected. Clear the FE flag by reading SCSR1 with FE set and then reading SCDR.

PF — Parity error flag

1 (set) – Incorrect parity detected.

0 (clear) – Parity correct.

PF is set if received data has incorrect parity. Clear PF by reading SCSR1 with PE set and then reading SCDR.

7.6.5 SCSR2 — SCI status register 2

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
SCI status 2 (SCSR2)	$0075	0	0	0	0	0	0	0	RAF	0000 0000

In the SCSR2 only bit 0 is used, to indicate receiver active. The other seven bits always read zero.

Bits [7:1] — Not implemented; always read zero

RAF — Receiver active flag (read only)

1 (set) – A character is being received.

0 (clear) – A character is not being received.

7.6.6 SCDRH, SCDRL — SCI data high/low registers

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
SCI data high (SCDRH)	$0076	R8	T8	0	0	0	0	0	0	undefined
SCI data low (SCDRL)	$0077	R7T7	R6T6	R5T5	R4T4	R3T3	R2T2	R1T1	R0T0	undefined

SCDRH/SCDRL is a parallel register that performs two functions. It is the receive data register when it is read, and the transmit data register when it is written. Reads access the receive data buffer and writes access the transmit data buffer. Data received or transmitted is double buffered.

If the SCI is being used with 7 or 8-bit data, only SCDRL needs to be accessed. Note that if 9-bit data 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.

R8 — Receiver bit 8

Ninth serial data bit received when SCI is configured for a nine data bit operation

T8 — Transmitter bit 8

Ninth serial data bit transmitted when SCI is configured for a nine data bit operation

Bits [5:0] — Not implemented; always read zero

R/T[7:0] — Receiver/transmitter data bits [7:0]

SCI data is double buffered in both directions.

7.7 Status flags and interrupts

The SCI transmitter has two status flags. These status flags can be read by software (polled) to tell when certain conditions exist. Alternatively, a local interrupt enable bit can be set to enable each of these status conditions to generate interrupt requests. Status flags are automatically set by hardware logic conditions, but must be cleared by software. This provides an interlock mechanism that enables logic to know when software has noticed the status indication. The software clearing sequence for these flags is automatic — functions that are normally performed in response to the status flags also satisfy the conditions of the clearing sequence.

TDRE and TC flags are normally set when the transmitter is first enabled (TE set to one). The TDRE flag indicates there is room in the transmit queue to store another data character in the transmit data register. The TIE bit is the local interrupt mask for TDRE. When TIE is zero, TDRE must be polled. When TIE and TDRE are one, an interrupt is requested.

The TC flag indicates the transmitter has completed the queue. The TCIE bit is the local interrupt mask for TC. When TCIE is zero, TC must be polled; when TCIE is one and TC is one, an interrupt is requested.

Writing a zero to TE requests that the transmitter stop when it can. The transmitter completes any transmission in progress before shutting down. Only an MCU reset can cause the transmitter to stop and shut down immediately. If TE is cleared when the transmitter is already idle, the pin reverts to its general purpose I/O function (synchronized to the bit-rate clock). If anything is being transmitted when TE is cleared, that character is completed before the pin reverts to general purpose I/O, but any other characters waiting in the transmit queue are lost. The TC and TDRE flags are set at the completion of this last character, even though TE has been disabled.

7.7.1 Receiver flags

The SCI receiver has seven status flags, three of which can generate interrupt requests. The status flags are set by the SCI logic in response to specific conditions in the receiver. These flags can be read (polled) at any time by software. Refer to Figure 7-3, which shows SCI interrupt arbitration.

When an overrun takes place, the new character is lost, and the character that was in its way in the parallel receive data register (RDR) is undisturbed. RDRF is set when a character has been received and transferred into the parallel RDR. The OR flag is set instead of RDRF if overrun occurs. A new character is ready to be transferred into the RDR before a previous character is read from the RDR.

The NF, FE and PF flags provide additional information about the character in the RDR, but do not generate interrupt requests.

The receiver active flag (RAF) indicates that the receiver is busy.

The last receiver status flag and interrupt source come from the IDLE flag. The RXD line is idle if it has constantly been at logic one for a full character time. The IDLE flag is set only after the RXD line has been busy and becomes idle. This prevents repeated interrupts for the time RXD remains idle.

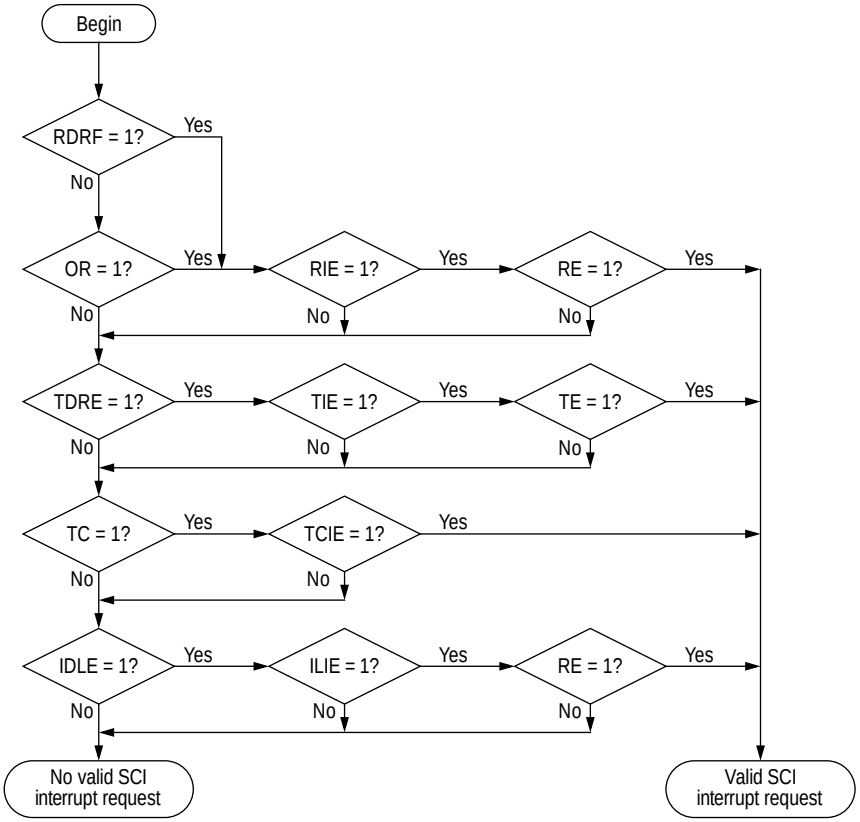


Figure 7-3 Interrupt source resolution within SCI

8
SERIAL PERIPHERAL INTERFACE

The serial peripheral interface (SPI), an independent serial communications subsystem, allows the MCU to communicate synchronously with peripheral devices, such as transistor-transistor logic (TTL) shift registers, liquid crystal (LCD) display drivers, analog-to-digital converter subsystems, and other microprocessors. The SPI is also capable of inter-processor communication in a multiple master system. The SPI system can be configured as either a master or a slave device, with data rates as high as one half of the E clock rate when configured as a master and as fast as the E clock rate when configured as a slave.

The SPI shares I/O with four of port D's pins and is enabled by SPE in the SPCR.

Pin	Alternative function
PD2	MISO
PD3	MOSI
PD4	SCK
PD5	$\overline{SS}$

8.1 Functional description

The central element in the SPI system is the block containing the shift register and the read data buffer (see Figure 8-1). The system is single buffered in the transmit direction and double buffered in the receive direction. This means that new data for transmission cannot be written to the shifter until the previous transfer is complete; however, received data is transferred into a parallel read data buffer so the shifter is free to accept a second serial character. As long as the first character is read out of the read data buffer before the next serial character is ready to be transferred, no overrun condition occurs. A single MCU register address is used for reading data from the read data buffer and for writing data to the shifter.

The SPI status block represents the SPI status functions (transfer complete, write collision, and mode fault) performed by the serial peripheral status register (SPSR). The SPI control block represents those functions that control the SPI system through the serial peripheral control register (SPCR).

8.2 SPI transfer formats

During an SPI transfer, data is simultaneously transmitted and received. A serial clock line synchronizes shifting and sampling of the information on the two serial data lines. A slave select line allows individual selection of a slave SPI device; slave devices that are not selected do not interfere with SPI bus activities. On a master SPI device, the select line can optionally be used to indicate a multiple master bus contention. Refer to Figure 8-2.

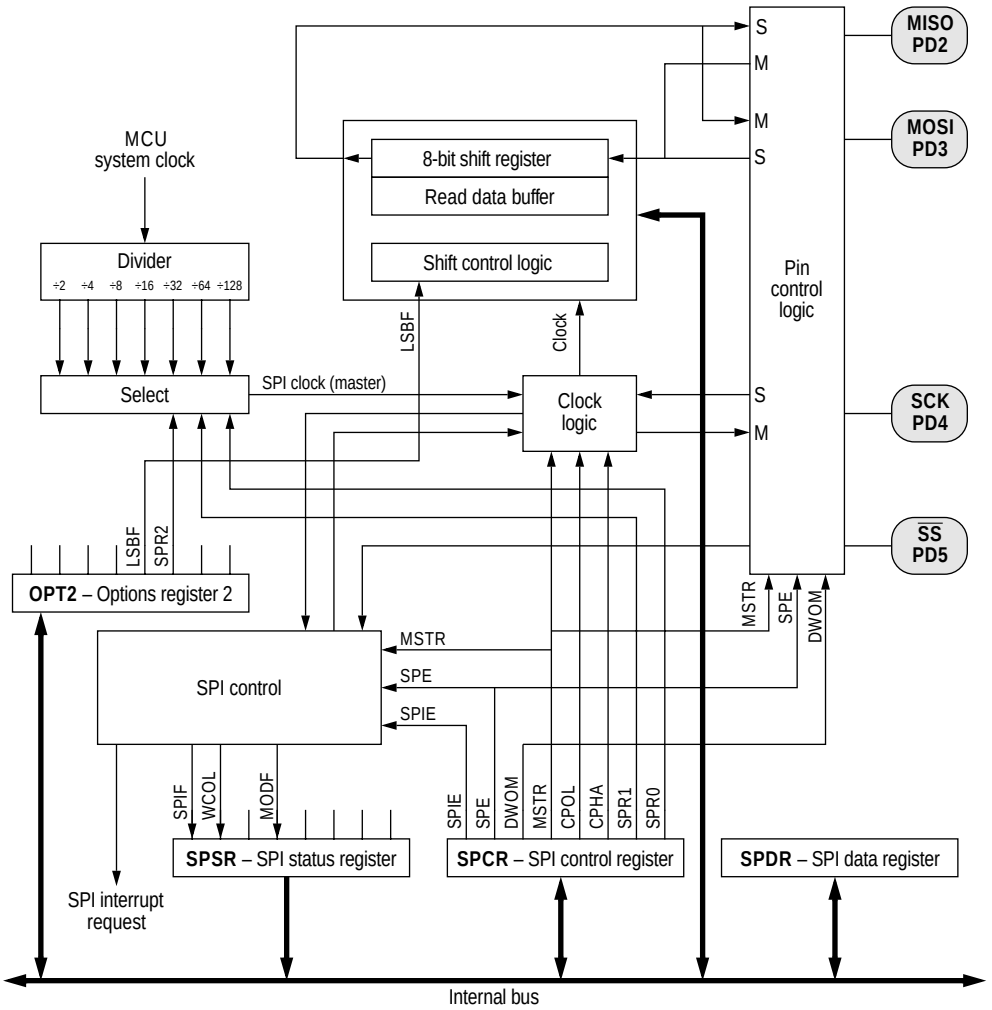
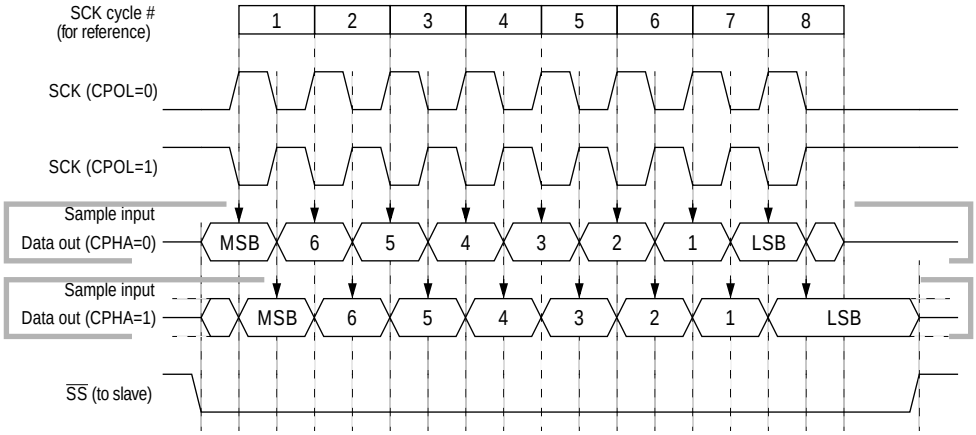


Figure 8-1 SPI block diagram



Note: this figure shows the LSBF=0 (default) case. If LSBF=1, data is transferred in the reverse order (LSB first).

Figure 8-2 SPI transfer format

8.2.1 Clock phase and polarity controls

Software can select one of four combinations of serial clock phase and polarity using two bits in the SPI control register (SPCR). The clock polarity is specified by the CPOL control bit, which selects an active high or active low clock, and has no significant effect on the transfer format. The clock phase (CPHA) control bit selects one of two different transfer formats. The clock phase and polarity should be identical for the master SPI device and the communicating slave device. In some cases, the phase and polarity are changed between transfers to allow a master device to communicate with peripheral slaves having different requirements.

When CPHA equals zero, the $\overline{SS}$ line must be deasserted and reasserted between each successive serial byte. Also, if the slave writes data to the SPI data register (SPDR) while $\overline{SS}$ is low, a write collision error results.

When CPHA equals one, the $\overline{SS}$ line can remain low between successive transfers.

8.3 SPI signals

The following paragraphs contain descriptions of the four SPI signals: master in slave out (MISO), master out slave in (MOSI), serial clock (SCK), and slave select ($\overline{SS}$).

Any SPI output line must have its corresponding data direction bit in DDRD register set. If the DDR bit is clear, that line is disconnected from the SPI logic and becomes a general-purpose input. All SPI input lines are forced to act as inputs regardless of the state of the corresponding DDR bits in DDRD register.

8.3.1 Master in slave out

MISO is one of two unidirectional serial data signals. It is an input to a master device and an output from a slave device. The MISO line of a slave device is placed in the high-impedance state if the slave device is not selected.

8.3.2 Master out slave in

The MOSI line is the second of the two unidirectional serial data signals. It is an output from a master device and an input to a slave device. The master device places data on the MOSI line a half-cycle before the clock edge that the slave device uses to latch the data.

8.3.3 Serial clock

SCK, an input to a slave device, is generated by the master device and synchronizes data movement in and out of the device through the MOSI and MISO lines. Master and slave devices are capable of exchanging a byte of information during a sequence of eight clock cycles.

There are four possible timing relationships that can be chosen by using control bits CPOL and CPHA in the serial peripheral control register (SPCR). Both master and slave devices must operate with the same timing. The SPI clock rate select bits, SPR[1:0], in the SPCR of the master device, select the clock rate. In a slave device, SPR[1:0] have no effect on the operation of the SPI.

8.3.4 Slave select

The slave select $\overline{SS}$ input of a slave device must be externally asserted before a master device can exchange data with the slave device. $\overline{SS}$ must be low before data transactions begin and must stay low for the duration of the transaction.

The $\overline{SS}$ line of the master must be held high. If it goes low, a mode fault error flag (MODF) is set in the serial peripheral status register (SPSR). To disable the mode fault circuit, write a one in bit 5 of the port D data direction register. This sets the $\overline{SS}$ pin to act as a general-purpose output, rather than a dedicated input to the slave select circuit, thus inhibiting the mode fault flag. The other three lines are dedicated to the SPI whenever the serial peripheral interface is on.

The state of the master and slave CPHA bits affects the operation of $\overline{SS}$. CPHA settings should be identical for master and slave. When CPHA = 0, the shift clock is the OR of $\overline{SS}$ with SCK. In this clock phase mode, $\overline{SS}$ must go high between successive characters in an SPI message. When CPHA = 1, $\overline{SS}$ can be left low between successive SPI characters. In cases where there is only one SPI slave MCU, its $\overline{SS}$ line can be tied to V_{SS} as long as only CPHA = 1 clock mode is used.

8.4 SPI system errors

Two kinds of system errors can be detected by the SPI system. The first type of error arises in a multiple-master system when more than one SPI device simultaneously tries to be a master. This error is called a mode fault. The second type of error, write collision, indicates that an attempt was made to write data to the SPDR while a transfer was in progress.

When the SPI system is configured as a master and the $\overline{SS}$ input line goes to active low, a mode fault error has occurred — usually because two devices have attempted to act as master at the same time. In the case where more than one device is concurrently configured as a master, there is a chance of contention between two pin drivers. For push-pull CMOS drivers, this contention can cause permanent damage. The mode fault detection circuitry attempts to protect the device by disabling the drivers. The MSTR control bit in the SPCR and all four DDRD control bits associated with the SPI are cleared and an interrupt is generated (subject to masking by the SPIE control bit and the I bit in the CCR).

Other precautions may need to be taken to prevent driver damage. If two devices are made masters at the same time, the mode fault detector does not help protect either one unless one of them selects the other as slave. The amount of damage possible depends on the length of time both devices attempt to act as master.

A write collision error occurs if the SPDR is written while a transfer is in progress. Because the SPDR is not double buffered in the transmit direction, writes to SPDR cause data to be written directly into the SPI shift register. Because this write corrupts any transfer in progress, a write collision error is generated. The transfer continues undisturbed, and the write data that caused the error is not written to the shifter.

A write collision is normally a slave error because a slave has no control over when a master initiates a transfer. A master knows when a transfer is in progress, so there is no reason for a master to generate a write-collision error, although the SPI logic can detect write collisions in both master and slave devices.

The SPI configuration determines the characteristics of a transfer in progress. For a master, a transfer begins when data is written to SPDR and ends when SPIF is set. For a slave with CPHA equal to zero, a transfer starts when $\overline{SS}$ goes low and ends when $\overline{SS}$ returns high. In this case, SPIF is set at the middle of the eighth SCK cycle when data is transferred from the shifter to the parallel data register, but the transfer is still in progress until $\overline{SS}$ goes high. For a slave with CPHA equal to one, transfer begins when the SCK line goes to its active level, which is the edge at the beginning of the first SCK cycle. The transfer ends when SPIF is set, for a slave in which CPHA=1.

8.5 SPI registers

The three SPI registers, SPCR, SPSR, and SPDR, provide control, status, and data storage functions. Refer to the following information for a description of how these registers are organized.

8.5.1 SPCR — SPI control register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
SPI control (SPCR)	$0028	SPIE	SPE	DWOM	MSTR	CPOL	CPHA	SPR1	SPR0	0000 01uu

This register can be read and written at any time.

SPIE — Serial peripheral interrupt enable

1 (set) – A hardware interrupt sequence is requested each time SPIF or MODF is set.

0 (clear) – SPI interrupts are inhibited.

Set the SPIE bit to a one to request a hardware interrupt sequence each time the SPIF or MODF status flag is set. SPI interrupts are inhibited if this bit is clear or if the I bit in the condition code register is one.

SPE — Serial peripheral system enable

1 (set) – Port D [5:2] is dedicated to the SPI.

0 (clear) – Port D has its default I/O functions and the clock generator is stopped.

When the SPE bit is set, the port D pins 2, 3, 4, and 5 are dedicated to the SPI functions and lose their general purpose I/O functions. When the SPI system is enabled and expects any of PD[4:2] to be inputs then those pins will be inputs regardless of the state of the associated DDRD bits. If any of PD[4:2] are expected to be outputs then those pins will be outputs only if the associated DDRD bits are set. However, if the SPI is in the master mode, DDD5 determines whether PD5 is an error detect input (DDD5 = 0) or a general-purpose output (DDD5 = 1).

DWOM — Port D wired-OR mode

1 (set) – Port D [5:2] buffers configured for open-drain outputs.

0 (clear) – Port D [5:2] buffers configured for normal CMOS outputs.

MSTR — Master mode select

1 (set) – Master mode

0 (clear) – Slave mode

CPOL — Clock polarity

1 (set) – SCK is active low.

0 (clear) – SCK is active high.

When the clock polarity bit is cleared and data is not being transferred, the SCK pin of the master device has a steady state low value. When CPOL is set, SCK idles high. Refer to Figure 8-2 and Section 8.2.1.

CPHA — Clock phase

The clock phase bit, in conjunction with the CPOL bit, controls the clock-data relationship between master and slave. The CPHA bit selects one of two different clocking protocols. Refer to Figure 8-2 and Section 8.2.1.

SPR1 and SPR0 — SPI clock rate selects

These two bits select the SPI clock rate, as shown in Table 8-1. Note that SPR2 is located in the OPT2 register, and that its state on reset is zero.

Table 8-1 SPI clock rates

SPR[2:0]	E clock divide ratio	SPI clock frequency (≡ baud rate) for: E = 4MHz
0 0 0	2	2.0 MHz
0 0 1	4	1.0 MHz
0 1 0	16	250 kHz
0 1 1	32	125 kHz
1 0 0	8	500 kHz
1 0 1	16	250 kHz
1 1 0	64	62.5 kHz
1 1 1	128	31.3 kHz

8.5.2 SPSR — SPI status register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
SPI status (SPSR)	$0029	SPIF	WCOL	0	MODF	0	0	0	0	0000 0000

This register can be read at any time, but writing to it has no effect.

SPIF — SPI interrupt complete flag

1 (set) – Data transfer to external device has been completed.

0 (clear) – No valid completion of data transfer.

SPIF is set upon completion of data transfer between the processor and the external device. If SPIF goes high, and if SPIE is set, a serial peripheral interrupt is generated. To clear the SPIF bit, read the SPSR with SPIF set, then access the SPDR. Unless SPSR is read (with SPIF set) first, attempts to write SPDR are inhibited.

WCOL — Write collision

1 (set) – Write collision.

0 (clear) – No write collision.

Clearing the WCOL bit is accomplished by reading the SPSR (with WCOL set) followed by an access of SPDR. Refer to Section 8.3.4 and Section 8.4.

MODF — Mode fault

1 (set) – Mode fault.

0 (clear) – No mode fault.

To clear the MODF bit, read the SPSR (with MODF set), then write to the SPCR. Refer to Section 8.3.4 and Section 8.4.

Bits [5, 3:0] — Not implemented; always read zero.

8.5.3 SPDR — SPI data register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
SPI data (SPDR)	$002A	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	undefined

The SPDR is used when transmitting or receiving data on the serial bus. Only a write to this register initiates transmission or reception of a byte, and this only occurs in the master device. At the completion of transferring a byte of data, the SPIF status bit is set in both the master and slave devices.

A read of the SPDR is actually a read of a buffer. To prevent an overrun and the loss of the byte that caused the overrun, the first SPIF must be cleared by the time a second transfer of data from the shift register to the read buffer is initiated.

SPI is double buffered in and single buffered out.

8.5.4 OPT2 — System configuration options register 2

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
System config. options 2 (OPT2)	$0038	LIRDV	CWOM	0	IRVNE	LSBF	SPR2	XDV1	XDV0	000x 0000

LIRDV — $\overline{\text{LIR}}$ driven (refer to Section 4)

1 (set) – Enable $\overline{\text{LIR}}$ drive high pulse.

0 (clear) – $\overline{\text{LIR}}$ not driven high on MODA/$\overline{\text{LIR}}$ pin.

CWOM — Port C wired-OR mode (refer to Section 6)

1 (set) – Port C outputs are open-drain.

0 (clear) – Port C operates normally.

Bit 5 — Not implemented; always reads zero.

IRVNE — Internal read visibility/not E (refer to Section 4)

1 (set) – Data from internal reads is driven out of the external data bus.

0 (clear) – No visibility of internal reads on external bus.

In **single chip mode** this bit determines whether the E clock drives out from the chip.

1 (set) – E pin is driven low.

0 (clear) – E clock is driven out from the chip.

LSBF — LSB first enable

1 (set) – SPI data is transferred LSB first.

0 (clear) – SPI data is transferred MSB first.

If this bit is set, data, which is usually transferred MSB first, is transferred LSB first. LSBF does not affect the position of the MSB and LSB in the data register. Reads and writes of the data register always have MSB in bit 7.

SPR2 — SPI clock rate select

When set, SPR2 adds a divide-by-4 prescaler to the SPI clock chain. With the two bits in the SPCR, this bit specifies the SPI clock rate. Refer to Table 8-1.

XDV[1, 0] — XOUT clock divide select (refer to Section 4)

These two bits control the frequency of the XCLK signal, which is output on the XOUT pin if enabled by the CLKX bit in CONFIG.

9
TIMING SYSTEM

The MC68HC11KW1 contains three 16-bit timers. Figure 9-1 provides a diagram of the entire timing system. The main timer, Timer 1, is described in the following paragraphs; refer to Section 9.2 and Section 9.3 for descriptions of Timer 2 and Timer 3.

9.1 Timer 1

Timer 1 is the standard M68HC11 timing system, composed of several clock divider chains. The main clock divider chain includes a 16-bit free-running counter, driven by a programmable prescaler.

The prescaler output divides the system clock by 1, 4, 8, or 16. Taps from this main clocking chain drive-circuitry generate the slower clocks used by the pulse accumulator, the real-time interrupt (RTI), and the computer operating properly (COP) watchdog subsystems, which are also described in this section. Refer to Figure 9-1.

All main timer system activities are referenced to this free-running counter. The counter begins incrementing from $0000 as the MCU comes out of reset, and continues to the maximum count, $FFFF. At the maximum count, the counter rolls over to $0000, sets an overflow flag and continues to increment. As long as the MCU is running in a normal operating mode, there is no way to reset, change or interrupt the counting. The capture/compare subsystem features three input capture channels, four output compare channels and one channel that can be selected to perform either input capture or output compare. Each of the three input capture functions has its own 16-bit input capture register (time capture latch) and each of the output compare functions has its own 16-bit compare register. All timer functions, including the timer overflow and RTI, have their own interrupt controls and separate interrupt vectors.

The pulse accumulator contains an 8-bit counter and edge select logic. The pulse accumulator can operate in either event counting mode or gated time accumulation mode. During event counting mode, the pulse accumulator's 8-bit counter increments when a specified edge is detected on an input signal. During gated time accumulation mode, an internal clock source increments the 8-bit counter while an input signal has a predetermined logic level.

The real-time interrupt (RTI) is a programmable periodic interrupt circuit that permits pacing the execution of software routines by selecting one of four interrupt rates.

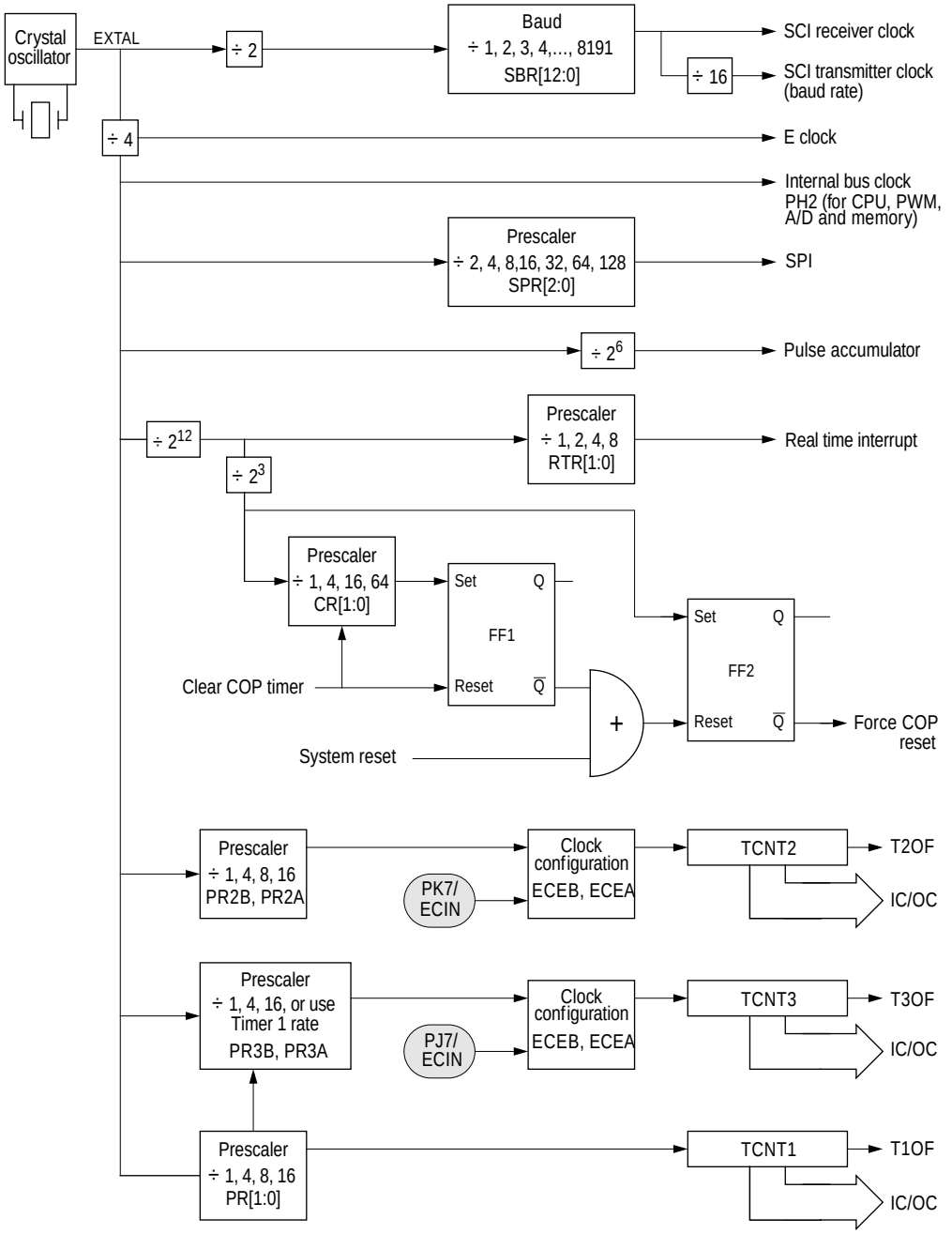


Figure 9-1 Timer clock divider chains

The COP watchdog clock input ($E/2^{15}$) is tapped off from the free-running counter chain. The COP automatically times out unless it is serviced within a specific time by a program reset sequence. If the COP is allowed to time out, a reset is generated, which drives the $\overline{\text{RESET}}$ pin low to reset the MCU and the external system. Refer to Table 9-1 for crystal related frequencies and periods.

Table 9-1 Timer 1 resolution and capacity

Control bits PR[1:0]	Clock: 16.0 MHz / 4.0 MHz / 250 ns	Clock: 4 E / E / 1/E	XTAL / E clock / Period
0 0	250 ns 16.384 ms	1/E 2^{16}/E	– resolution – overflow
0 1	1.0 µs 65.536 ms	4/E 2^{18}/E	– resolution – overflow
1 0	2.0 µs 131.07 ms	8/E 2^{19}/E	– resolution – overflow
1 1	4.0 µs 262.14 ms	16/E 2^{20}/E	– resolution – overflow

9.1.1 Timer 1 structure

The functions of Timer 1 share I/O with the pins of port A as follows:

Pin	Alternative function
PA0	IC3
PA1	IC2
PA2	IC1
PA3	OC5 and/or OC1, or IC4
PA4	OC4 and/or OC1
PA5	OC3 and/or OC1
PA6	OC2 and/or OC1
PA7	PAI and/or OC1

Figure 9-3 shows the Timer 1 capture/compare system block diagram. The port A pin control block includes logic for timer functions and for general-purpose I/O. For pins PA3, PA2, PA1 and PA0, this block contains both the edge-detection logic and the control logic that enables the selection of which edge triggers an input capture. The digital level on PA[3:0] can be read at any time (read PORTA register), even if the pin is being used for the input capture function. Pins PA[6:3] are used either for general-purpose I/O, or as output compare pins. When one of these pins is being used for an output compare function, it cannot be written directly as if it were a general-purpose output. Each of the output compare functions (OC[5:2]) is related to one of the port A output pins. Output

compare 1 (OC1) has extra control logic, allowing it optional control of any combination of the PA[7:3] pins. The PA7 pin can be used as a general-purpose I/O pin, as an input to the pulse accumulator or as an OC1 output pin.

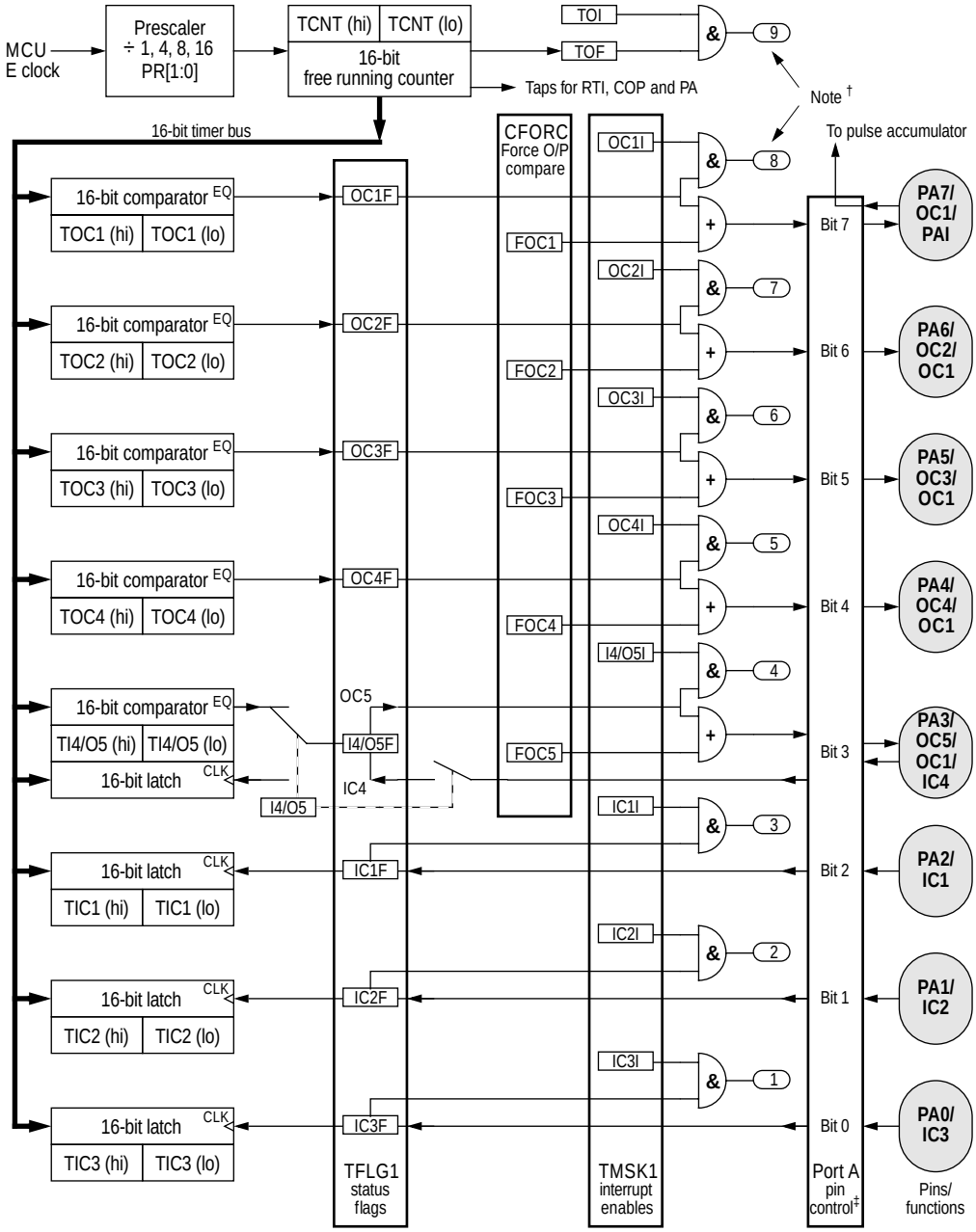
9.1.2 Input capture

The input capture function records the time an external event occurs by latching the value of the free-running counter when a selected edge is detected at the associated timer input pin. Software can store latched values and use them to compute the periodicity and duration of events. For example, by storing the times of successive edges of an incoming signal, software can determine the period and pulse width of a signal. To measure period, two successive edges of the same polarity are captured. To measure pulse width, two alternate polarity edges are captured.

In most cases, input capture edges are asynchronous with respect to the internal timer counter, which is clocked relative to an internal clock (PH2). These asynchronous capture requests are synchronized with PH2 so that latching occurs on the opposite half cycle of PH2 from when the timer counter is being incremented. This synchronization process introduces a delay from when the edge occurs to when the counter value is detected. Because these delays cancel out when the time between two edges is being measured, the delay can be ignored. When an input capture is being used with an output compare, there is a similar delay between the actual compare point and when the output pin changes state.

The control and status bits that implement the input capture functions are contained in the PACTL, TCTL2, TMSK1, and TFLG1 registers.

To configure port A bit 3 as an input capture, clear the DDA3 bit of the DDRA register. Note that this bit is cleared out of reset. To enable PA3 as the fourth input capture, set the I4/O5 bit in the PACTL register. Otherwise, PA3 is configured as a fifth output compare out of reset, with bit I4/O5 being cleared. If the DDA3 bit is set (configuring PA3 as an output), and IC4 is enabled, then writes to PA3 cause edges on the pin to result in input captures. Writing to TI4/O5 has no effect when the TI4/O5 register is acting as IC4.



† Interrupt requests 1–9 (these are further qualified by the I-bit in the CCR)

‡ Port A pin actions are controlled by OC1M, OC1D, PACTL, TCTL1 and TCTL2 registers

Figure 9-2 Timer 1 capture/compare block diagram

9.1.2.1 TCTL2 — Timer control register 2

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Timer control 2 (TCTL2)	$0021	EDG4B	EDG4A	EDG1B	EDG1A	EDG2B	EDG2A	EDG3B	EDG3A	0000 0000

Use the control bits of this register to program input capture functions to detect a particular edge polarity on the corresponding timer input pin. Each of the input capture functions can be independently configured to detect rising edges only, falling edges only, any edge (rising or falling), or to disable the input capture function. The input capture functions operate independently of each other and can capture the same TCNT value if the input edges are detected within the same timer count cycle.

EDGxB and EDGxA — Input capture edge control

EDGxB	EDGxA	Configuration
0	0	ICx disabled
0	1	ICx captures on rising edges only
1	0	ICx captures on falling edges only
1	1	ICx captures on any edge

There are four pairs of these bits. Each pair is cleared by reset and must be encoded to configure the corresponding input capture edge detector circuit. IC4 functions only if the I4/O5 bit in the PACTL register is set.

9.1.2.2 TIC1–TIC3 — Timer input capture registers

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Timer input capture 1 (TIC1) high	$0010	(bit 15)	(14)	(13)	(12)	(11)	(10)	(9)	(bit 8)	not affected
Timer input capture 1 (TIC1) low	$0011	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	not affected
Timer input capture 2 (TIC2) high	$0012	(bit 15)	(14)	(13)	(12)	(11)	(10)	(9)	(bit 8)	not affected
Timer input capture 2 (TIC2) low	$0013	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	not affected
Timer input capture 3 (TIC3) high	$0014	(bit 15)	(14)	(13)	(12)	(11)	(10)	(9)	(bit 8)	not affected
Timer input capture 3 (TIC3) low	$0015	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	not affected

When an edge has been detected and synchronized, the 16-bit free-running counter value is transferred into the input capture register pair as a single 16-bit parallel transfer. Timer counter value captures and timer counter incrementing occur on opposite half-cycles of the phase 2 clock so that the count value is stable whenever a capture occurs. Input capture values can be read from a pair of 8-bit read-only registers. A read of the high-order byte of an input capture register pair inhibits a new capture transfer for one bus cycle. If a double-byte read instruction, such as LDD, is used to read the captured value, coherency is assured. When a new input capture occurs immediately after a high-order byte read, transfer is delayed for an additional cycle but the value is not lost.

The TICx registers are not affected by reset.

9.1.2.3 TI4/O5 — Timer input capture 4/output compare 5 register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Capture 4/compare 5 (TI4/O5) high	$001E	(bit 15)	(14)	(13)	(12)	(11)	(10)	(9)	(bit 8)	1111 1111
Capture 4/compare 5 (TI4/O5) low	$001F	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	1111 1111

Use TI4/O5 as either an input capture register or an output compare register, depending on the function chosen for the PA3 pin. To enable it as an input capture pin, set the I4/O5 bit in the pulse accumulator control register (PACTL) to logic level one. To use it as an output compare register, set the I4/O5 bit to a logic level zero. Refer to Section 9.6.1.

The TI4/O5 register pair resets to ones ($FFFF).

9.1.3 Output compare

Use the output compare (OC) function to program an action to occur at a specific time — when the 16-bit counter reaches a specified value. For each of the five output compare functions, there is a separate 16-bit compare register and a dedicated 16-bit comparator. The value in the compare register is compared to the value of the free-running counter on every bus cycle. When the compare register matches the counter value, an output compare status flag is set. The flag can be used to initiate the automatic actions for that output compare function.

To produce a pulse of a specific duration, write a value to the output compare register that represents the time the leading edge of the pulse is to occur. The output compare circuit is configured to set the appropriate output either high or low, depending on the polarity of the pulse being produced. After a match occurs, the output compare register is reprogrammed to change the output pin back to its inactive level at the next match. A value representing the width of the pulse is added to the original value, and then written to the output compare register. Because the pin state changes occur at specific values of the free-running counter, the pulse width can be controlled accurately at the resolution of the free-running counter, independent of software latency. To generate an output signal of a specific frequency and duty cycle, repeat this pulse-generating procedure.

There are four 16-bit read/write output compare registers: TOC1, TOC2, TOC3, and TOC4, and the TI4/O5 register, which functions under software control as either IC4 or OC5. Each of the OC registers is set to $FFFF on reset. A value written to an OC register is compared to the free-running counter value during each E clock cycle. If a match is found, the particular output compare flag is set in timer interrupt flag register 1 (TFLG1). If that particular interrupt is enabled in the timer interrupt mask register 1 (TMSK1), an interrupt is generated. In addition to an interrupt, a specified action can be initiated at one or more timer output pins. For OC[5:2], the pin action is controlled by pairs of bits (OMx and OLx) in the TCTL1 register. The output action is taken on each successful compare, regardless of whether or not the OCxF flag in the TFLG1 register was previously cleared.

OC1 is different from the other output compares in that a successful OC1 compare can affect any or all five of the OC pins. The OC1 output action taken when a match is found is controlled by two 8-bit registers with three bits unimplemented: the output compare 1 mask register, OC1M, and the output compare 1 data register, OC1D. OC1M specifies which port A outputs are to be used, and OC1D specifies what data is placed on these port pins.

9.1.3.1 TOC1–TOC4 — Timer output compare registers

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Timer output compare 1 (TOC1) high	$0016	(bit 15)	(14)	(13)	(12)	(11)	(10)	(9)	(bit 8)	1111 1111
Timer output compare 1 (TOC1) low	$0017	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	1111 1111
Timer output compare 2 (TOC2) high	$0018	(bit 15)	(14)	(13)	(12)	(11)	(10)	(9)	(bit 8)	1111 1111
Timer output compare 2 (TOC2) low	$0019	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	1111 1111
Timer output compare 3 (TOC3) high	$001A	(bit 15)	(14)	(13)	(12)	(11)	(10)	(9)	(bit 8)	1111 1111
Timer output compare 3 (TOC3) low	$001B	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	1111 1111
Timer output compare 4 (TOC4) high	$001C	(bit 15)	(14)	(13)	(12)	(11)	(10)	(9)	(bit 8)	1111 1111
Timer output compare 4 (TOC4) low	$001D	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	1111 1111

All output compare registers are 16-bit read-write. Each is initialized to $FFFF at reset. If an output compare register is not used for an output compare function, it can be used as a storage location. A write to the high-order byte of an output compare register pair inhibits the output compare function for one bus cycle. This inhibition prevents inappropriate subsequent comparisons. Coherency requires a complete 16-bit read or write. However, if coherency is not needed, byte accesses can be used.

For output compare functions, write a comparison value to output compare registers TOC1–TOC4 and TI4/O5. When TCNT value matches the comparison value, specified pin actions occur.

9.1.3.2 CFORC — Timer compare force register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Timer compare force (CFORC)	$000B	FOC1	FOC2	FOC3	FOC4	FOC5	0	0	0	0000 0000

The CFORC register allows forced early compares. FOC[1:5] correspond to the five output compares. These bits are set for each output compare that is to be forced. The action taken as a result of a forced compare is the same as if there were a match between the OCx register and the free-running counter, except that the corresponding interrupt status flag bits are not set. The forced channels trigger their programmed pin actions to occur at the next timer count transition after the write to CFORC.

The CFORC bits should not normally be used on an output compare function that is programmed to toggle its output on a successful compare, because a normal compare occurring immediately before or after the force would produce a double toggle. This may be undesirable if it happens quickly, since the resulting output pulse would be very short.

FOC[1:5] — Force output compares

1 (set) – A forced output compare action will occur on the specified pin.

0 (clear) – No action.

Bits [2:0] — Not implemented; always read zero

9.1.3.3 OC1M — Output compare 1 mask register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Output compare 1 mask (OC1M)	$000C	OC1M7	OC1M6	OC1M5	OC1M4	OC1M3	0	0	0	0000 0000

Use OC1M with OC1 to specify the bits of port A that are affected by a successful OC1 compare. The bits of the OC1M register correspond to PA7–PA3.

OC1M[7:3] — Output compare masks for OC1

1 (set) – OC1 is configured to control the corresponding pin of port A.

0 (clear) – OC1 will not affect the corresponding port A pin.

Bits [2:0] — Not implemented; always read zero.

9.1.3.4 OC1D — Output compare 1 data register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Output compare 1 data (OC1D)	$000D	OC1D7	OC1D6	OC1D5	OC1D4	OC1D3	0	0	0	0000 0000

Use this register with OC1 to specify the data that is to be written to the affected pin of port A after a successful OC1 compare. When a successful OC1 compare occurs, a data bit in OC1D is written to the corresponding pin of port A for each bit that is set in OC1M.

OC1D[7:3] — Output compare data for OC1

If OC1Mx is set, data in OC1Dx is output to port A pin x on successful OC1 compares.

Bits [2:0] — Not implemented; always read zero

9.1.3.5 TCNT — Timer counter register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Timer count (TCNT) high	$000E	(bit 15)	(14)	(13)	(12)	(11)	(10)	(9)	(bit 8)	0000 0000
Timer count (TCNT) low	$000F	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	0000 0000

The 16-bit read-only TCNT register contains the prescaled value of the 16-bit timer. A full counter read addresses the more significant byte (MSB) first. A read of this address causes the less significant byte (LSB) to be latched into a buffer for the next CPU cycle so that a double-byte read returns the full 16-bit state of the counter at the time of the MSB read cycle.

TCNT resets to $0000.

9.1.3.6 TCTL1 — Timer control register 1

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Timer control 1 (TCTL1)	$0020	OM2	OL2	OM3	OL3	OM4	OL4	OM5	OL5	0000 0000

The bits of this register specify the action taken as a result of a successful OCx compare.

OM[2:5] — Output mode
OL[2:5] — Output level

OMx	OLx	Action taken on successful compare
0	0	Timer disconnected from OCx pin logic
0	1	Toggle OCx output line
1	0	Clear OCx output line to 0
1	1	Set OCx output line to 1

These control bit pairs are encoded to specify the action taken after a successful OCx compare. OC5 functions only if the I4/O5 bit in the PACTL register is clear.

9.1.3.7 TMSK1 — Timer interrupt mask register 1

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Timer interrupt mask 1 (TMSK1)	$0022	OC1I	OC2I	OC3I	OC4I	I4/O5I	IC1I	IC2I	IC3I	0000 0000

Use this 8-bit register to enable or inhibit the timer input capture and output compare interrupts.

Note: Bits in TMSK1 correspond bit for bit with flag bits in TFLG1. Ones in TMSK1 enable the corresponding interrupt sources.

OC1I–OC4I — Output compare x interrupt enable

1 (set) – OCx interrupt is enabled.

0 (clear) – OCx interrupt is disabled.

If the OCxI enable bit is set when the OCxF flag bit is set, a hardware interrupt sequence is requested.

I4/O5I — Input capture 4/output compare 5 interrupt enable

1 (set) – IC4/OC5 interrupt is enabled.

0 (clear) – IC4/OC5 interrupt is disabled.

When I4/O5 in PACTL is set, I4/O5I is the input capture 4 interrupt enable bit.
When I4/O5 in PACTL is zero, I4/O5I is the output compare 5 interrupt enable bit.

IC1I–IC3I — Input capture x interrupt enable

1 (set) – ICx interrupt is enabled.

0 (clear) – ICx interrupt is disabled.

If the ICxI enable bit is set when the ICxF flag bit is set, a hardware interrupt sequence is requested.

9.1.3.8 TFLG1 — Timer interrupt flag register 1

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Timer interrupt flag 1 (TFLG1)	$0023	OC1F	OC2F	OC3F	OC4F	I4/O5F	IC1F	IC2F	IC3F	0000 0000

Bits in this register indicate when timer system events have occurred. Coupled with the bits of TMSK1, the bits of TFLG1 allow the timer subsystem to operate in either a polled or interrupt driven system. Clear flags by writing a one to the corresponding bit position(s).

Note: Bits in TFLG1 correspond bit for bit with flag bits in TMSK1. Ones in TMSK1 enable the corresponding interrupt sources.

OC1F–OC4F — Output compare x flag

1 (set) – Counter has reached the preset output compare x value.

0 (clear) – Counter has not reached the preset output compare x value.

These flags are set each time the counter matches the corresponding output compare x values.

I4/O5F — Input capture 4/output compare 5 flag

Set by IC4 or OC5, depending on the function enabled by I4/O5 bit in PACTL

IC1F–IC3F — Input capture x flag

1 (set) – Selected edge has been detected on corresponding port pin.

0 (clear) – Selected edge has not been detected on corresponding port pin.

These flags are set each time a selected active edge is detected on the ICx input line

9.1.3.9 TMSK2 — Timer interrupt mask register 2

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Timer interrupt mask 2 (TMSK2)	$0024	TOI	RTII	PAOVI	PAII	0	0	PR1	PR0	0000 0000

Use this 8-bit register to enable or inhibit timer overflow and real-time interrupts. The timer prescaler control bits are included in this register.

Note: Bits [7:4] in TMSK2 correspond bit for bit with the flag bits in TFLG2. Ones in bits [7:4] of TMSK2 enable the corresponding interrupt sources.

TOI — Timer overflow interrupt enable

1 (set) – Timer overflow interrupt requested when TOF is set.

0 (clear) – TOF interrupts disabled.

RTII — Real-time interrupt enable (refer to Section 9.4)

PAOVI — Pulse accumulator overflow interrupt enable (refer to Section 9.6.3)

PAII — Pulse accumulator input edge interrupt enable (refer to Section 9.6.3)

PR[1:0] — Timer prescaler select

PR[1:0]	Prescaler
0 0	1
0 1	4
1 0	8
1 1	16

These bits are used to select the prescaler divide-by ratio. In normal modes, PR[1:0] can only be written once, and the write must be within 64 cycles after reset. See Table 9-1 for specific timing values. These two bits also specify the number of divide-by-two stages that are to be inserted between the E-clock and the timer free-running counter of Timer 3. This enables Timer 1-Timer 3 synchronization. This can, however, be overridden if a different prescale is required for Timer 3, by writing to the Timer 3 prescale bits in the register TCTL6. See Section 9.3.5. The default state is that the Timer 1 prescale rate is used for Timer 3.

9.1.3.10 TFLG2 — Timer interrupt flag register 2

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Timer interrupt flag 2 (TFLG2)	$0025	TOF	RTIF	PAOVF	PAIF	0	0	0	0	0000 0000

Bits in this register indicate when certain timer system events have occurred. Coupled with the four high-order bits of TMSK2, the bits of TFLG2 allow the timer subsystem to operate in either a polled or interrupt driven system. Clear flags by writing a one to the corresponding bit position(s).

Note: Bits in TFLG2 correspond bit for bit with flag bits in TMSK2. Ones in TMSK2 enable the corresponding interrupt sources.

TOF — Timer overflow interrupt flag

1 (set) – TCNT has overflowed from $FFFF to $0000.

0 (clear) – No timer overflow has occurred.

RTIF — Real time (periodic) interrupt flag (refer to Section 9.4)

PAOVF — Pulse accumulator overflow interrupt flag (refer to Section 9.6)

PAIF — Pulse accumulator input edge interrupt flag (refer to Section 9.6.)

Bits [3:0] — Not implemented; always read zero

9.2 Timer 2

Timer 2 comprises a 4-stage prescaler and a 16-bit counter. It has three associated 16-bit output compare registers along with a software-programmable input capture or output compare register.

The functions of Timer 2 share I/O with the pins of port J as follows:

Pin	Alternative function
PJ3	ECIN
PJ4	OC1
PJ5	OC2
PJ6	OC3
PJ7	C4

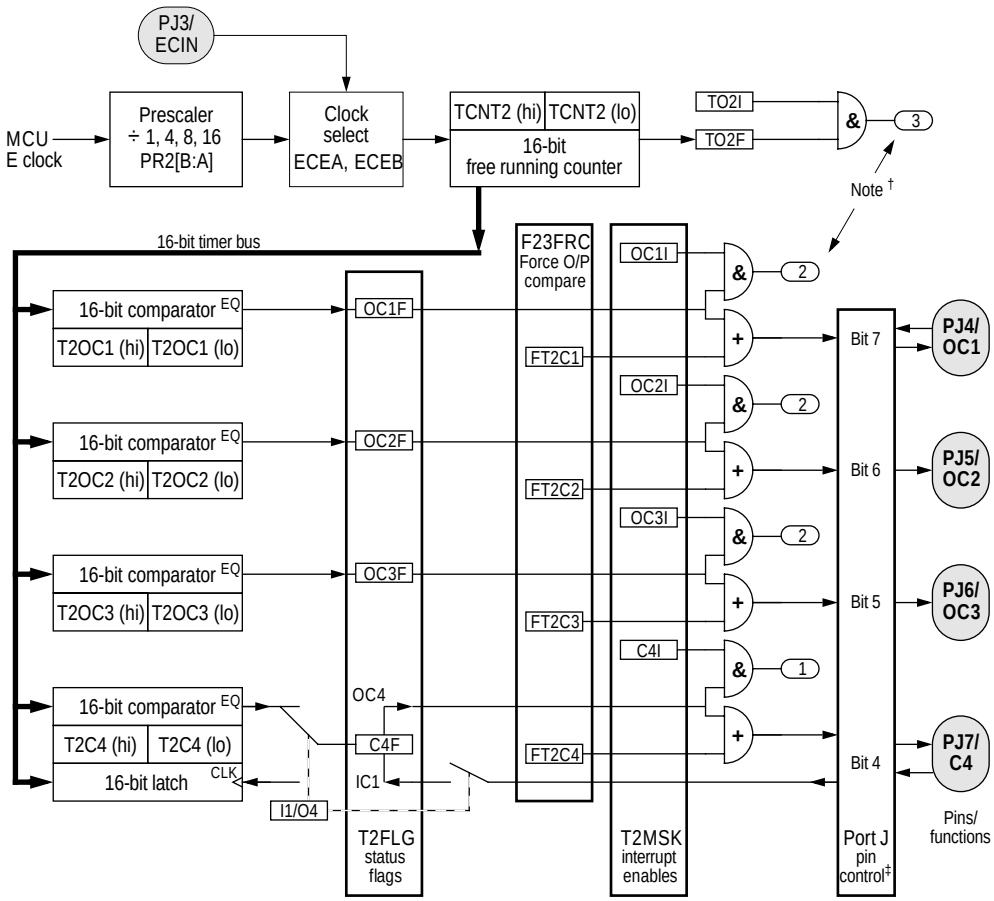
The Timer 2 prescaler is a 4 stage divider with the E clock as its input. Prescaling factors of 1, 4, 8 or 16 can be selected by the P2RA and P2RB bits in the TCTL4 register. Timer 2 also offers an

event counting mode of operation, in which the counter is clocked by an external source. In this case, the prescaler is bypassed and the external signal is used to clock the Timer 2 counter directly.

The input capture and output compare signals (OC1, OC2, OC3 and I1/OC4) are interfaced through port J pins [7:4]. If the timer is configured for event counting mode, then port J pin 3 is used as the input for the external clock. Refer to Figure 9-1 and Figure 9-3.

The Timer 2 counter can be read at any time, and can be reset to $0000 by writing any value to it (in normal modes). In addition, the counter can be stopped and reset by setting the T2STP bit in the TCTL4 register (see Section 9.2.8).

Timer 2 can be switched to event counting mode by writing to the ECEA and ECEB bits in the TCTL4 register. When switching to event counting mode, the counter is not reset to zero, but carries on from the state it was in at the time of switching. Immediately before the switch it was sourced by the internal E clock, so the first 16-stage count cycle has an offset equal to the number of E clock cycles after reset before the first write to TCTL4. The T2STP bit can be used to reset the counter to zero, if required.



† Interrupt requests 1, 2 & 3 (these are further qualified by the I-bit in the CCR).

‡ Port J pin actions are controlled by the TCTL3 and TCTL4 registers.

Figure 9-3 Timer 2 capture/compare block diagram

9.2.1 Output compare

There are three dedicated output compare registers associated with Timer 2 (T20C[3:1]). In addition, a fourth register (T2C4) is used either for a fourth output compare function or for an input capture function, depending on the state of the I1/O4 bit in the TCTL4 register.

The output compare function operates in a similar fashion to that on Timer 1; a value written to an output compare register is compared to the free running counter value during each clock cycle. If a match is found, the appropriate output compare flag is set in the interrupt flag register (T2FLG). An interrupt is then generated if that particular interrupt is enabled in the interrupt mask register (T2MSK). Unlike Timer 1, which has separate interrupt requests for each interrupt, Timer 2 output compare interrupts 1,2 and 3 are associated with a single interrupt sequence. This means that flag polling is required if more than one interrupt is enabled. Refer to Figure 9-3.

In addition to an interrupt, a successful output compare can trigger a specified action at the associated output port pins PJ [7:4]. The type of action taken is defined by bits in timer control register TCTL3. This action may be forced by setting the appropriate bit in the F23FRC register (see Section 9.2.3).

9.2.2 Input capture

When configured as an input capture register, the 16-bit T2C4 register is used to latch the value of the counter when a selected transition at pin PJ7 is detected. The type of transition which triggers the capture is defined by the EDGA and EDGB bits in the TCTL4 register.

9.2.3 F23FRC — Compare force register for Timers 2 and 3.

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Compare force for Timers 2 and 3 (F23FRC)	$0031	FT3C1	FT3C2	FT3C3	FT3C4	FT2C1	FT2C3	FT2C3	FT2C4	0000 0000

The F23FRC register allows forced early compares for Timers 2 and 3. Bits [7:4] correspond to the four output compares of Timer 3, and bits [3:0] correspond to those of Timer 2. These bits are set for each output compare that is to be forced. The action taken as a result of a forced compare is the same as if there were a match between the OCx register and the free-running counter, except that the corresponding interrupt status flag bits are not set. The forced channels trigger their programmed pin actions to occur at the next timer count transition after the write to F23FRC.

The F23FRC bits should not normally be used on an output compare function that is programmed to toggle its output on a successful compare, because a normal compare occurring immediately before or after the force would produce a double toggle. This may be undesirable if it happens quickly, since the resulting output pulse would be very short.

FT3C[1:4] and FT2C[1:4] — Force output compares

1 (set) – A forced output compare action will occur on the appropriate pin.

0 (clear) – No action.

9.2.4 T2C4 — Timer 2 channel 4 register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Timer 2 channel 4 (T2C4) high	$008A	(bit 15)	(14)	(13)	(12)	(11)	(10)	(9)	(bit 8)	1111 1111
Timer 2 channel 4 (T2C4) low	$008B	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	1111 1111

Use T2C4 as either an input capture register or an output compare register, depending on the function chosen for the PJ7 pin. To enable it as an input capture pin, set the I1/O4 bit in the timer control register 4 (TCTL4). To use it as an output compare pin, clear the I1/O4 bit. Refer to Section 9.2.8.

The T2C4 register pair resets to ones ($FFFF).

9.2.5 T2OC1–T2OC3 — Timer 2 output compare registers

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Timer 2 output compare 1 (T2OC1) high	$0084	(bit 15)	(14)	(13)	(12)	(11)	(10)	(9)	(bit 8)	1111 1111
Timer 2 output compare 1 (T2OC1) low	$0085	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	1111 1111
Timer 2 output compare 2 (T2OC2) high	$0086	(bit 15)	(14)	(13)	(12)	(11)	(10)	(9)	(bit 8)	1111 1111
Timer 2 output compare 2 (T2OC2) low	$0087	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	1111 1111
Timer 2 output compare 3 (T2OC3) high	$0088	(bit 15)	(14)	(13)	(12)	(11)	(10)	(9)	(bit 8)	1111 1111
Timer 2 output compare 3 (T2OC3) low	$0089	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	1111 1111

These three output compare registers are 16-bit read-write. Each is initialized to $FFFF at reset. If an output compare register is not used for an output compare function, it can be used as a storage location. A write to the high-order byte of an output compare register pair inhibits the output compare function for one bus cycle. This inhibition prevents inappropriate subsequent comparisons. Coherency requires a complete 16-bit read or write. However, if coherency is not needed, byte accesses can be used.

For output compare functions, write a comparison value to output compare registers T2OC1–T2OC3 and TI1/O4. When TCNT2 value matches the comparison value, the specified pin actions occur.

9.2.6 TCNT2 — Timer 2 counter register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Timer 2 count (TCNT2) high	$0082	(bit 15)	(14)	(13)	(12)	(11)	(10)	(9)	(bit 8)	0000 0000
Timer 2 count (TCNT2) low	$0083	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	0000 0000

This 16-bit register can be read at any time. In normal modes (SMOD = 0), writing any value to the counter causes it to be reset to $0000. In special modes (SMOD = 1), any write to the most significant byte (MSB) causes the counter to be preset to $FFF8. This preset capability is intended for factory testing only. The counter can be stopped and reset by writing to the T2STP bit in the TCTL4 register (see Section 9.2.8).

The TCNT2 register contains the value of the 16-bit timer. A full counter read addresses the most significant byte (MSB) first. A read of this address causes the less significant byte (LSB) to be latched into a buffer for the next CPU cycle so that a double-byte read returns the full 16-bit state of the counter at the time of the MSB read cycle.

TCNT2 resets to $0000.

9.2.7 TCTL3 — Timer control register 3 (Timer 2)

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Timer control register 3 (TCTL3)	$0080	OM1	OL1	OM2	OL2	OM3	OL3	OM4	OL4	0000 0000

The bits of this register specify the action taken as a result of a successful Timer 2 OCx compare.

OM[1:4] — Output mode
OL[1:4] — Output level

OMx	OLx	Action taken on successful compare
0	0	Timer disconnected from OCx pin logic
0	1	Toggle OCx output line
1	0	Clear OCx output line to 0
1	1	Set OCx output line to 1

These control bit pairs are encoded to specify the action taken after a successful OCx compare. OC4 functions only if the I1/O4 bit in the TCTL4 register is clear.

9.2.8 TCTL4 — Timer control register 4 (Timer 2)

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Timer control register 4 (TCTL4)	$0081	EDGB	EDGA	PR2B	PR2A	ECEB	ECEA	T2STP	I1/04	0000 0000

EDGB and EDGA — Input capture edge control

This pair of bits configures the input capture edge detector circuits for IC1. IC1 functions only if the I1/O4 bit is set.

EDGB	EDGA	Configuration
0	0	IC1 disabled
0	1	IC1 captures on rising edges only
1	0	IC1 captures on falling edges only
1	1	IC1 captures on any edge

Note: The maximum frequency of the input clock must be less than E/2 when counting on one edge, and less that E/4 when counting on both edges.

PR2A and PR2B — Timer 2 prescaler select

These bits are used to select the prescaler divide-by ratio for Timer 2. They can only be written to once after reset.

PR2B	PR2A	Prescaler
0	0	1
0	1	4
1	0	8
1	1	16

ECEB and ECEA — Event counter edge control

These control bits configure the input clock source for the Timer 2 counter. They can be written to only once after reset.

ECEB	ECEA	Configuration
0	0	Timer 2 uses internal clock and prescaler
0	1	Count on rising edges of external clock only
1	0	Count on falling edges of external clock only
1	1	Count on any edge of external clock

T2STP — Stop Timer 2 counter

1 (set) – Timer 2 counter and prescaler are stopped and the counter is reset to $0000.

0 (clear) – Timer 2 counter operates normally.

I1/O4 — Input capture 1/output compare 4

1 (set) – Input capture 1 function is enabled (no OC4)

0 (clear) – Output compare 4 function is enabled (no IC1)

9.2.9 T2MSK — Timer 2 interrupt mask register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Timer 2 interrupt mask (T2MSK)	$008C	OC1I	OC2I	OC3I	C4I	TO2I	0	0	0	0000 0000

Use this 8-bit register to enable or inhibit the timer input capture and output compare interrupts.

Note: Bits in T2MSK correspond bit for bit with flag bits in T2FLG. Ones in T2MSK enable the corresponding interrupt sources.

OC1I–OC3I — Output compare x interrupt enable

1 (set) – OC interrupt is enabled.

0 (clear) – OC interrupt is disabled.

If an OCxI enable bit is set when its associated OCxF flag bit is set, a hardware interrupt sequence is requested. All three interrupt enable bits are associated with a single Timer 2 output compare interrupt sequence; any successful output compare causes such an interrupt unless the corresponding OCxI bit is clear. Therefore, flag polling is required unless all but one of the interrupts have been disabled.

C4I— Input capture 1/output compare 4 interrupt enable

1 (set) – IC1/OC4 interrupt is enabled.

0 (clear) – IC1/OC4 interrupt is disabled.

When I1/O4 in TCTL4 is set, C4I is the input capture 1 interrupt enable bit.
When I1/O4 in TCTL4 is zero, C4I is the output compare 4 interrupt enable bit.

TO2I — Timer 2 overflow interrupt enable

1 (set) – Timer 2 overflow interrupt requested when T2OF is set.

0 (clear) – T2OF interrupts disabled.

Bits [2:0] — Not implemented; always read zero.

9.2.10 T2FLG — Timer 2 interrupt flag register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Timer 2 interrupt flag (T2FLG)	$008D	OC1F	OC2F	OC3F	C4F	TO2F	0	0	0	0000 0000

Bits in this register indicate when timer system events have occurred. Clear flags by writing a one to the corresponding bit position(s).

Note: Bits in T2FLG correspond bit for bit with flag bits in T2MSK. Ones in T2MSK enable the corresponding interrupt sources.

OC1F–OC3F — Output compare x flag

1 (set) – Counter has reached the preset output compare x value.

0 (clear) – Counter has not reached the preset output compare x value.

These flags are set each time the counter matches the corresponding output compare x values.

C4F — Input capture 1/output compare 4 flag

This flag is set by IC1 or OC4, depending on the function enabled by I1/O4 bit in TCTL4. If C4 is configured as an input capture channel, then:

1 (set) – Selected edge has been detected on pin PJ7

0 (clear) – Selected edge has not been detected on pin PJ7.

TO2F — Timer 2 overflow flag

1 (set) – TCNT2 has overflowed from $FFFF to $0000.

0 (clear) – No Timer 2 overflow has occurred.

Bits [2:0] — Not implemented; always read zero.

9.3 Timer 3

Timer 3 is very similar in operation to Timer 2. Its functions share I/O with port K pins as follows:

Pin	Alternative function
PK3	ECIN
PK4	OC1
PK5	OC2
PK6	OC3
PK7	C4

When not in event counting mode, Timer 3 can be driven either by the E clock divided by 1, 4, or 16, or by the Timer 1 clock rate, as shown in Figure 9-1. See also Section 9.1.3.9.

Another difference between Timers 2 and 3 is that Timer 3 has only two interrupt request sequences associated with it (IC1/OC4/OC[3:1] and timer overflow), whereas Timer 2 has three (OC[3:1], IC1/OC4 and timer overflow). See Section 5 and Section 9.3.5.

Apart from these two differences, Timer 3 operates in the same way as Timer 2. The control and data registers associated with Timer 3 are described in the following paragraphs.

Note: Timer 3 output compares can be forced using bits in the F23FRC register, described in Section 9.2.3.

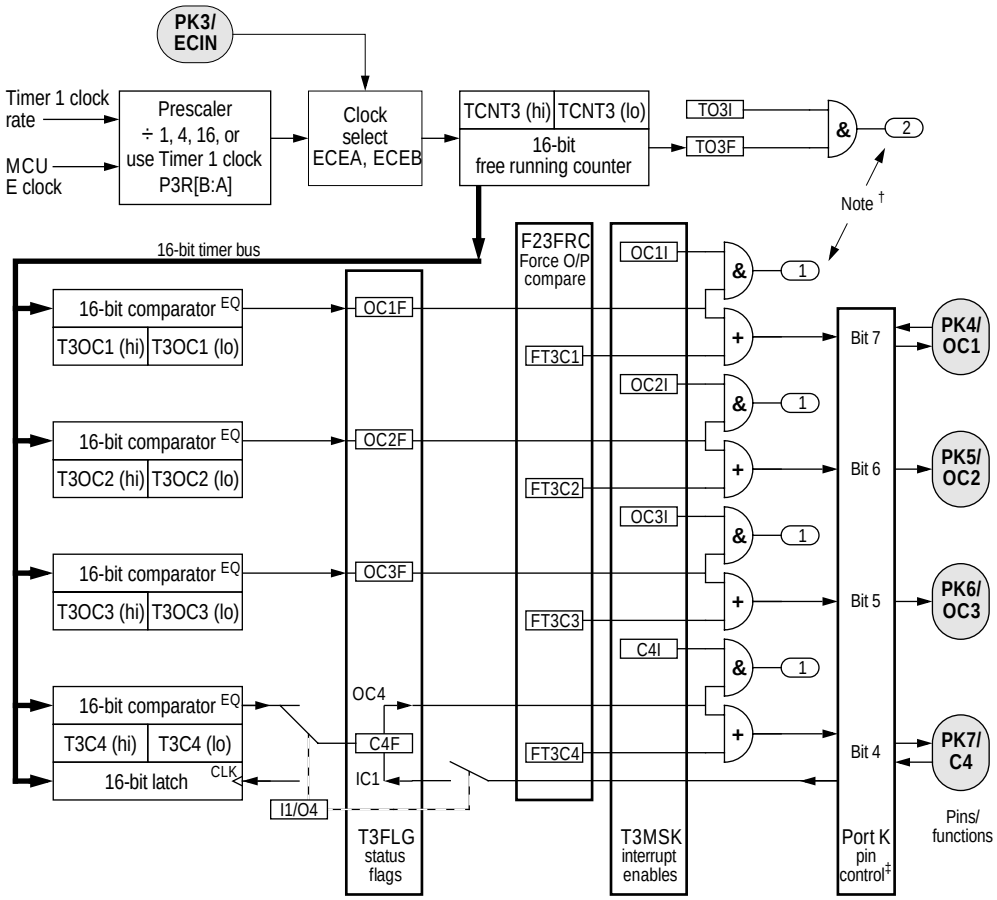
9

9.3.1 T3C4 — Timer 3 channel 4 register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Timer 3 channel 4 (T2C4) high	$009A	(bit 15)	(14)	(13)	(12)	(11)	(10)	(9)	(bit 8)	1111 1111
Timer 3 channel 4(T2C4) low	$009B	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	1111 1111

Use T3C4 as either an input capture register or an output compare register, depending on the function chosen for the PK7 pin. To enable it as an input capture pin, set the I1/O4 bit in the timer control register 4 (TCTL6). To use it as an output compare pin, clear the I1/O4 bit. Refer to Section 9.3.5.

The T3C4 register pair resets to ones ($FFFF).



† Interrupt requests 1 and 2 (these are further qualified by the I-bit in the CCR).

‡ Port K pin actions are controlled by the TCTL5 and TCTL6 registers.

Figure 9-4 Timer 3 capture/compare block diagram

9.3.2 T3OC1–T3OC3 — Timer 3 output compare registers

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Timer 3 output compare 1 (T3OC1) high	$0094	(bit 15)	(14)	(13)	(12)	(11)	(10)	(9)	(bit 8)	1111 1111
Timer 3 output compare 1 (T3OC1) low	$0095	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	1111 1111
Timer 3 output compare 2 (T3OC2) high	$0096	(bit 15)	(14)	(13)	(12)	(11)	(10)	(9)	(bit 8)	1111 1111
Timer 3 output compare 2 (T3OC2) low	$0097	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	1111 1111
Timer 3 output compare 3 (T3OC3) high	$0098	(bit 15)	(14)	(13)	(12)	(11)	(10)	(9)	(bit 8)	1111 1111
Timer 3 output compare 3 (T3OC3) low	$0099	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	1111 1111

These three output compare registers are 16-bit read-write. Each is initialized to $FFFF at reset. If an output compare register is not used for an output compare function, it can be used as a storage location. A write to the high-order byte of an output compare register pair inhibits the output compare function for one bus cycle. This inhibition prevents inappropriate subsequent comparisons. Coherency requires a complete 16-bit read or write. However, if coherency is not needed, byte accesses can be used.

For output compare functions, write a comparison value to output compare registers T3OC1–T3OC3 and TI1/O4. When the TCNT3 value matches the comparison value, specified pin actions occur.

9

9.3.3 TCNT3 — Timer 3 counter register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Timer 3 count (TCNT3) high	$0092	(bit 15)	(14)	(13)	(12)	(11)	(10)	(9)	(bit 8)	0000 0000
Timer 3 count (TCNT3) low	$0093	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	0000 0000

This register can be read at any time. In normal modes (SMOD = 0), writing any value to the counter causes it to be rest to $0000. In special modes (SMOD = 1), any write to the most significant byte (MSB) causes the counter to be preset to $FFF8.This preset capability is intended for factory testing only. The counter can be stopped and reset by writing to the T3STP bit in the TCTL6 register (see Section 9.3.5).

The 16-bit read-only TCNT3 register contains the value of the 16-bit timer. A full counter read addresses the most significant byte (MSB) first. A read of this address causes the less significant byte (LSB) to be latched into a buffer for the next CPU cycle so that a double-byte read returns the full 16-bit state of the counter at the time of the MSB read cycle.

TCNT3 resets to $0000.

9.3.4 TCTL5 — Timer control register 5 (Timer 3)

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Timer control register 5 (TCTL5)	$0090	OM1	OL1	OM2	OL2	OM3	OL3	OM4	OL4	0000 0000

The bits of this register specify the action taken as a result of a successful Timer 3 OCx compare.

OM[1:4] — Output mode
OL[1:4] — Output level

OMx	OLx	Action taken on successful compare
0	0	Timer disconnected from OCx pin logic
0	1	Toggle OCx output line
1	0	Clear OCx output line to 0
1	1	Set OCx output line to 1

These control bit pairs are encoded to specify the action taken after a successful OCx compare. OC4 functions only if the I1/O4 bit in the TCTL6 register is clear.

9.3.5 TCTL6 — Timer control register 6 (Timer 3)

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Timer control register 6 (TCTL6)	$0091	EDGB	EDGA	PR3B	PR3A	ECEB	ECEA	T3STP	I1/04	0000 0000

EDGB and EDGA — Input capture edge control

This pair of bits configures the input capture edge detector circuits for IC1. IC1 functions only if the I1/O4 bit is set.

EDGB	EDGA	Configuration
0	0	IC1 disabled
0	1	IC1 captures on rising edges only
1	0	IC1 captures on falling edges only
1	1	IC1 captures on any edge

Note: The maximum frequency of the input clock must be less than E/2 when counting on one edge, and less that E/4 when counting on both edges.

PR3A and PR3B — Timer 3 prescaler select

These bits are used to select the prescaler divide-by ratio for Timer 3. They can only be written to once after reset. If PR3B and PR3A are both cleared, then Timer 3 is synchronized to the prescaled Timer 1 rate, which is determined by the PR1 and PR0 bits in the TMSK2 register. See Section 9.1.3.9 and Figure 9-1.

PR3B	PR3A	Prescaler
0	0	Use Timer 1 rate
0	1	1
1	0	4
1	1	16

ECEB and ECEA — Event counter edge control

These control bits configure the input clock source for the Timer 3 counter. They can be written to only once after reset.

ECEB	ECEA	Configuration
0	0	Timer 2 uses internal clock and prescaler
0	1	Count on rising edges of external clock only
1	0	Count on falling edges of external clock only
1	1	Count on any edge of external clock

T3STP — Stop Timer 3 counter

1 (set) – Timer 3 counter and prescaler are stopped and the counter is reset to $0000.

0 (clear) – Timer 3 counter operates normally.

I1/O4 — Input capture 1/output compare 4

1 (set) – Input capture 1 function is enabled (no OC4)

0 (clear) – Output compare 4 function is enabled (no IC1)

9.3.6 T3MSK — Timer 3 interrupt mask register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Timer 3 interrupt mask (T2MSK)	$009C	OC1I	OC2I	OC3I	C4I	TO2I	0	0	0	0000 0000

Use this 8-bit register to enable or inhibit the timer input capture and output compare interrupts.

Note: Bits in T3MSK correspond bit for bit with flag bits in T3FLG. Ones in T3MSK enable the corresponding interrupt sources.

OC1I–OC3I — Output compare x interrupt enable

1 (set) – OCx interrupt is enabled.

0 (clear) – OCx interrupt is disabled.

If an OCxI enable bit is set when its associated OCxF flag bit is set, a hardware interrupt sequence is requested. These three bits and the C4I bit are associated with a single Timer 3 output compare interrupt sequence; any successful output compare or input capture causes such an interrupt unless the corresponding OCxI or C4I bit is clear. Therefore flag polling is required if more than one interrupt is enabled.

C4I— Input capture 1/output compare 4 interrupt enable

1 (set) – IC1/OC4 interrupt is enabled.

0 (clear) – IC1/OC4 interrupt is disabled.

When I1/O4 in TCTL4 is set, C4I is the input capture 1 interrupt enable bit.
When I1/O4 in TCTL4 is zero, C4I is the output compare 4 interrupt enable bit.

TO3I — Timer 3 overflow interrupt enable

1 (set) – Timer 3 overflow interrupt requested when T3OF is set.

0 (clear) – T3OF interrupts disabled.

Bits [2:0] — Not implemented; always read zero.

9.3.7 T3FLG — Timer 3 interrupt flag register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Timer 3 interrupt flag (T3FLG)	$009D	OC1F	OC2F	OC3F	C4F	TO2F	0	0	0	0000 0000

Bits in this register indicate when timer system events have occurred. Clear flags by writing a one to the corresponding bit position(s).

Note: Bits in T3FLG correspond bit for bit with flag bits in T3MSK. Ones in T3MSK enable the corresponding interrupt sources.

OC1F–OC3F — Output compare x flag

1 (set) – Counter has reached the preset output compare x value.

0 (clear) – Counter has not reached the preset output compare x value.

These flags are set each time the counter matches the corresponding output compare x values.

C4F — Input capture 1/output compare 4flag

This flag is set by IC1 or OC4, depending on the function enabled by the I1/O4 bit in TCTL6. If C4 is configured as an input capture channel, then:

1 (set) – Selected edge has been detected on pin PK7

0 (clear) – Selected edge has not been detected on pin PK7.

TO3F — Timer 3 overflow flag

1 (set) – TCNT3 has overflowed from $FFFF to $0000.

0 (clear) – No Timer 3 overflow has occurred.

Bits [2:0] — Not implemented; always read zero.

9.4 Real-time interrupt

The real-time interrupt (RTI) feature, used to generate hardware interrupts at a fixed periodic rate, is controlled and configured by two bits (RTR1 and RTR0) in the pulse accumulator control (PACTL) register. The RTII bit in the TMSK3 register enables the interrupt capability. The four different rates available are a product of the MCU oscillator frequency and the value of bits RTR[1:0]. Refer to Table 9-2, which shows the periodic real-time interrupt rates.

Table 9-2 RTI periodic rates

RTR[1:0]	E = 4MHz	E = xMHz
0 0	1.02ms	2^{12}/E
0 1	2.05ms	2^{13}/E
1 0	4.09 ms	2^{14}/E
1 1	8.19 ms	2^{15}/E

The clock source for the RTI function is a free-running clock that cannot be stopped or interrupted except by reset. This clock causes the time between successive RTI timeouts to be a constant that is independent of the software latency associated with flag clearing and service. For this reason, an RTI period starts from the previous timeout, not from when RTIF is cleared.

Every timeout causes the RTIF bit in TFLG2 to be set, and if RTII is set, an interrupt request is generated. After reset, one entire RTI period elapses before the RTIF flag is set for the first time. Refer to the TMSK2, TFLG2, and PACTL registers.

9.4.1 TMSK2 — Timer interrupt mask register 2

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Timer interrupt mask 2 (TMSK2)	$0024	TOI	RTII	PAOVI	PAII	0	0	PR1	PR0	0000 0000

This register contains the real-time interrupt enable bit.

Note: Bits [7:4] in TMSK2 correspond bit for bit with flag bits in TFLG2. Ones in bits [7:4] TMSK2 enable the corresponding interrupt sources.

TOI — Timer overflow interrupt enable (refer to Section 9.1.3.9)

RTII — Real-time interrupt enable

1 (set) – Real time interrupt requested when RTIF is set.

0 (clear) – Real time interrupts disabled.

PAOVI — Pulse accumulator overflow interrupt enable (refer to Section 9.6)

PAII — Pulse accumulator input edge (refer to Section 9.6)

PR[1:0] — Timer prescaler select (refer to Section 9.1.3.9)

9.4.2 TFLG2 — Timer interrupt flag register 2

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Timer interrupt flag 2 (TFLG2)	$0025	TOF	RTIF	PAOVF	PAIF	0	0	0	0	0000 0000

Bits of this register indicate the occurrence of timer system events. Coupled with the four high-order bits of TMSK2, the bits of TFLG2 allow the timer subsystem to operate in either a polled or interrupt driven system. Clear flags by writing a one to the corresponding bit position(s).

Note: Bits in TFLG2 correspond bit for bit with flag bits in TMSK2. Ones in TMSK2 enable the corresponding interrupt sources.

TOF — Timer overflow interrupt flag

1 (set) – The timer has overflowed, from $FFFF to $0000.

0 (clear) – No timer overflow has occurred.

RTIF — Real-time interrupt flag

1 (set) – RTI period has elapsed.

0 (clear) – RTI flag has been cleared.

The RTIF status bit is automatically set to one at the end of every RTI period.

PAOVF — Pulse accumulator overflow interrupt flag (refer to Section 9.6)

PAIF — Pulse accumulator input edge interrupt flag (refer to Section 9.6)

Bits [3:0] — Not implemented; always read zero

9.4.3 PACTL — Pulse accumulator control register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Pulse accumulator control (PACTL)	$0026	0	PAEN	PAMOD	PEDGE	0	I4/O5	RTR1	RTR0	0000 0000

Bits RTR[1:0] of this register select the rate for the RTI system. The remaining bits control the pulse accumulator and IC4/OC5 functions.

Bits 7, 3 — Not implemented; always read zero

PAEN — Pulse accumulator system enable (refer to Section 9.6)

PAMOD — Pulse accumulator mode (refer to Section 9.6)

PEDGE — Pulse accumulator edge control (refer to Section 9.6)

I4/O5 — Input capture 4/output compare (refer to Section 9.6)

RTR[1:0] — RTI interrupt rate select

These two bits determine the rate at which the RTI system requests interrupts. The RTI system is driven by an $E/2^{12}$ clock rate that is compensated so it is independent of the timer prescaler. These two control bits select an additional division factor. Refer to Table 9-2.

9.5 Computer operating properly watchdog function

The clocking chain for the COP function, tapped off from the main timer divider chain, is only superficially related to the main timer system. The CR[1:0] bits in the OPTION register and the NOCOP bit in the CONFIG register determine the status of the COP function. One additional register, COPRST, is used to arm and clear the COP watchdog reset system. Refer to Section 5 for a more detailed discussion of the COP function.

9.6 Pulse accumulator

The MC68HC11KW1 has an 8-bit counter that can be configured to operate either as a simple event counter, or for gated time accumulation, depending on the state of the PAMOD bit in the PACTL register. Refer to the pulse accumulator block diagram, Figure 9-5.

In the event counting mode, the 8-bit counter is clocked to increasing values by an external pin. The maximum clocking rate for the external event counting mode is the E clock divided by two. In gated time accumulation mode, a free-running E clock ÷ 64 signal drives the 8-bit counter, but only

while the external PAI pin is activated. Refer to Table 9-3. The pulse accumulator counter can be read or written at any time.

Table 9-3 Pulse accumulator timing

Crystal frequency	E clock	Cycle time	64/E	PACNT overflow
16.0 MHz	4.0 MHz	250 ns	16.0 µs	4.096 ms

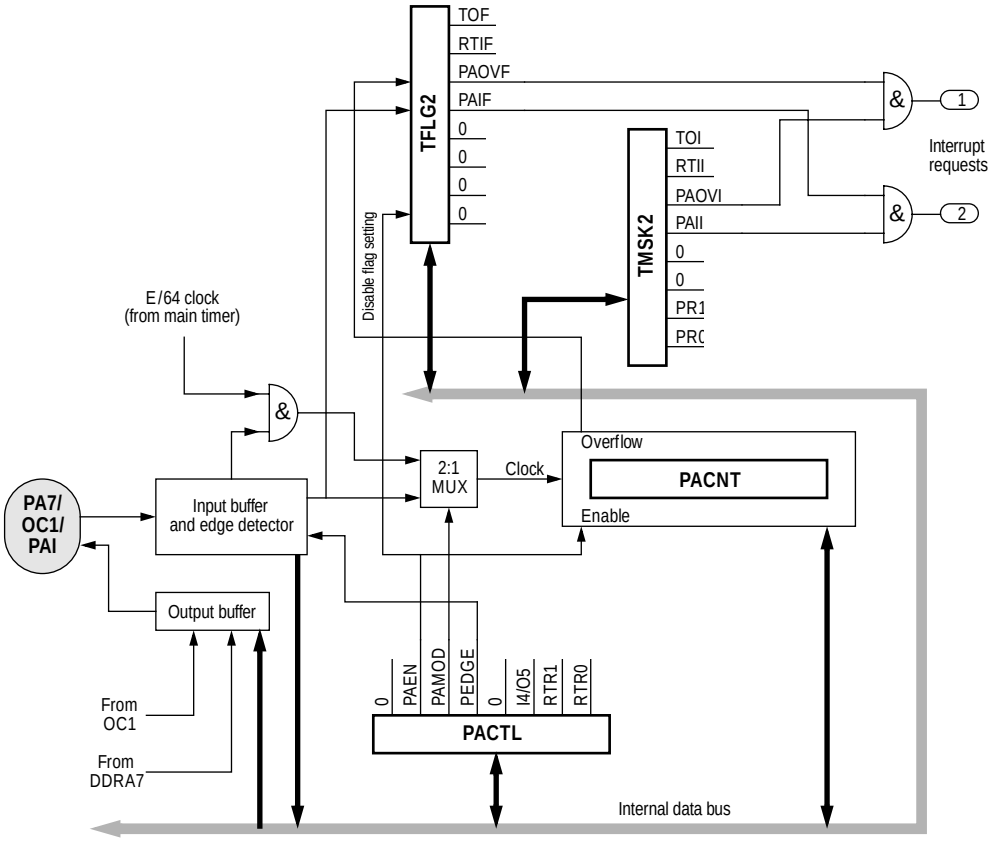


Figure 9-5 Pulse accumulator block diagram

Pulse accumulator control bits are also located within two timer registers, TMSK2 and TFLG2, as described in the following paragraphs.

9.6.1 PACTL — Pulse accumulator control register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Pulse accumulator control (PACTL)	$0026	0	PAEN	PAMOD	PEDGE	0	I4/O5	RTR1	RTR0	0000 0000

Four of this register's bits control an 8-bit pulse accumulator system. Another bit enables either the OC5 function or the IC4 function, while two other bits select the rate for the real-time interrupt system.

Bits [7, 3] — Not implemented; always read zero

PAEN — Pulse accumulator system enable

1 (set) – Pulse accumulator enabled.

0 (clear) – Pulse accumulator disabled.

PAMOD — Pulse accumulator mode

1 (set) – Gated time accumulation mode.

0 (clear) – Event counter mode.

PEDGE — Pulse accumulator edge control

This bit has different meanings depending on the state of the PAMOD bit, as shown:

PAMOD	PEDGE	Action of clock
0	0	PAI falling edge increments the counter.
0	1	PAI rising edge increments the counter.
1	0	A zero on PAI inhibits counting.
1	1	A one on PAI inhibits counting.

I4/O5 — Input capture 4/output compare 5

1 (set) – Input capture 4 function is enabled (no OC5).

0 (clear) – Output compare 5 function is enabled (no IC4).

RTR[1:0] — RTI interrupt rate selects (refer to Section 9.4)

9.6.2 PACNT — Pulse accumulator count register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Pulse accumulator count (PACNT)	$0027	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	undefined

This 8-bit read/write register contains the count of external input events at the PAI input, or the accumulated count. In gated time accumulation mode, PACNT is readable even if PAI is not active. The counter is not affected by reset and can be read or written at any time. Counting is synchronized to the internal PH2 clock so that incrementing and reading occur during opposite half cycles.

9.6.3 Pulse accumulator status and interrupt bits

The pulse accumulator control bits, PAOVI and PAII, PAOVF, and PAIF are located within timer registers TMSK2 and TFLG2.

9.6.3.1 TMSK2 — Timer interrupt mask 2 register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Timer interrupt mask 2 (TMSK2)	$0024	TOI	RTII	PAOVI	PAII	0	0	PR1	PR0	0000 0000

9.6.3.2 TFLG2 — Timer interrupt flag 2 register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Timer interrupt flag 2 (TFLG2)	$0025	TOF	RTIF	PAOVF	PAIF	0	0	0	0	0000 0000

PAOVI and PAOVF — Pulse accumulator interrupt enable and overflow flag

The PAOVF status bit is set each time the pulse accumulator count rolls over from $FF to $00. To clear this status bit, write a one in the corresponding data bit position (bit 5) of the TFLG2 register. The PAOVI control bit allows configuring the pulse accumulator overflow for polled or interrupt-driven operation and does not affect the state of PAOVF. When PAOVI is zero, pulse accumulator overflow interrupts are inhibited, and the system operates in a polled mode, which requires that PAOVF be polled by user software to determine when an overflow has occurred. When the PAOVI control bit is set, a hardware interrupt request is generated each time PAOVF is set. Before leaving the interrupt service routine, software must clear PAOVF by writing to the TFLG2 register.

PAII and PAIF — Pulse accumulator input edge interrupt enable and flag

The PAIF status bit is automatically set each time a selected edge is detected at the PA7/PAI/OC1 pin. To clear this status bit, write to the TFLG2 register with a one in the corresponding data bit position (bit 4). The PAII control bit allows configuring the pulse accumulator input edge detect for polled or interrupt-driven operation but does not affect setting or clearing the PAIF bit. When PAII is zero, pulse accumulator input interrupts are inhibited, and the system operates in a polled mode. In this mode, the PAIF bit must be polled by user software to determine when an edge has occurred. When the PAII control bit is set, a hardware interrupt request is generated each time PAIF is set. Before leaving the interrupt service routine, software must clear PAIF by writing to the TFLG register.

9.7 Pulse-width modulation (PWM) timer

The PWM timer subsystem provides up to four 8-bit pulse-width modulated waveforms on the port H pins. Channel pairs can be concatenated to create 16-bit PWM outputs. Three clock sources (A, B, and S) and a flexible clock select scheme give the PWM a wide range of frequencies.

Pin	Alternative function
PH0	PW1
PH1	PW2
PH2	PW3
PH3	PW4

Four control registers configure the PWM outputs — PWCLK, PWPOL, PWSCAL, and PWEN. The PWCLK register selects the prescale value for the PWM clock sources and enables the 16-bit PWM functions. The PWPOL register determines each channel's polarity and selects the clock source for each channel. The PWSCAL register derives a user-scaled clock based on the A clock source, and the PWEN register enables the PWM channels.

Each channel also has a separate 8-bit counter, period register, and duty cycle register. The period and duty cycle registers are double buffered so that if they are changed while the channel is enabled, the change does not take effect until the counter rolls over or the channel is disabled. A new period or duty cycle can be forced into effect immediately by writing to the period or duty cycle register and then writing to the counter.

With PWMs configured for 8-bit mode and E equal to 4 MHz, PWM signals can be produced from 40 kHz (1% duty cycle resolution) to less than 10 cycles per second (approximately 0.4% duty cycle resolution). By configuring the PWMs for 16-bit mode with E equal to 4 MHz, PWM periods greater than one minute are possible.

In 16-bit mode, duty cycle resolution of up to 15 parts per million can be achieved (at a PWM frequency of 60 Hz). In the same system, a PWM frequency of 1 kHz corresponds to a duty cycle resolution of 0.025%.

9.7.1 PWM timer block diagram

Figure 9-6 shows the block diagram of the PWM timer subsystem. Three different clock sources are selectable and provide inputs to the control registers. Each of the four channels has a counter, a period register, and a duty register. The waveform output is the result of a match between the period register (PWPERx) and the value in the counter (PWCNTx). The duty register (PWDTYx) changes the state of the output during the period to determine the duty cycle.

9.7.2 PWCLK — PWM clock prescaler and 16-bit select register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Pulse width clock select (PWCLK)	$0060	CON34	CON12	PCKA2	PCKA1	0	PCKB3	PCKB2	PCKB1	0000 0000

This register contains bits for selecting the 16-bit PWM options and the prescaler values for the clocks.

9.7.2.1 16-bit PWM function

The PWCLK register contains two control bits, each of which is used to concatenate a pair of PWM channels into one 16-bit channel. Channels 3 and 4 are concatenated with the CON34 bit, and channels 1 and 2 are concatenated with the CON12 bit.

When the 16-bit concatenated mode is selected, the clock source is determined by the low order channel. Channel 2 is the low order channel when channels 1 and 2 are concatenated. Channel 4 is the low order channel when channels 3 and 4 are concatenated. The pins associated with channels 1 and 3 can be used for general-purpose I/O when 16-bit PWM mode is selected.

Channel 1 registers are the high order byte of the double-byte channel when channels 1 and 2 are concatenated. Channel 3 registers are the high order byte of the double-byte channel when channels 3 and 4 are concatenated. Reads of the high order byte cause the low order byte to be latched for one cycle to guarantee that double byte reads are accurate. Writes to the low byte of the counter cause reset of the entire counter. Writes to the upper bytes of the counter have no effect.

CON34 — Concatenate channels 3 and 4

1 (set) – Channels 3 and 4 are concatenated into one 16-bit PWM channel.

0 (clear) – Channels 3 and 4 are separate 8-bit PWMs.

When concatenated, channel 3 is the high-order byte and the channel 4 pin (PH3) is the output.

CON12 — Concatenate Channels 1 and 2

1 (set) – Channels 1 and 2 are concatenated into one 16-bit PWM channel.

0 (clear) – Channels 1 and 2 are separate 8-bit PWMs.

When concatenated, channel 1 is the high-order byte and the channel 2 pin (PH1) is the output.

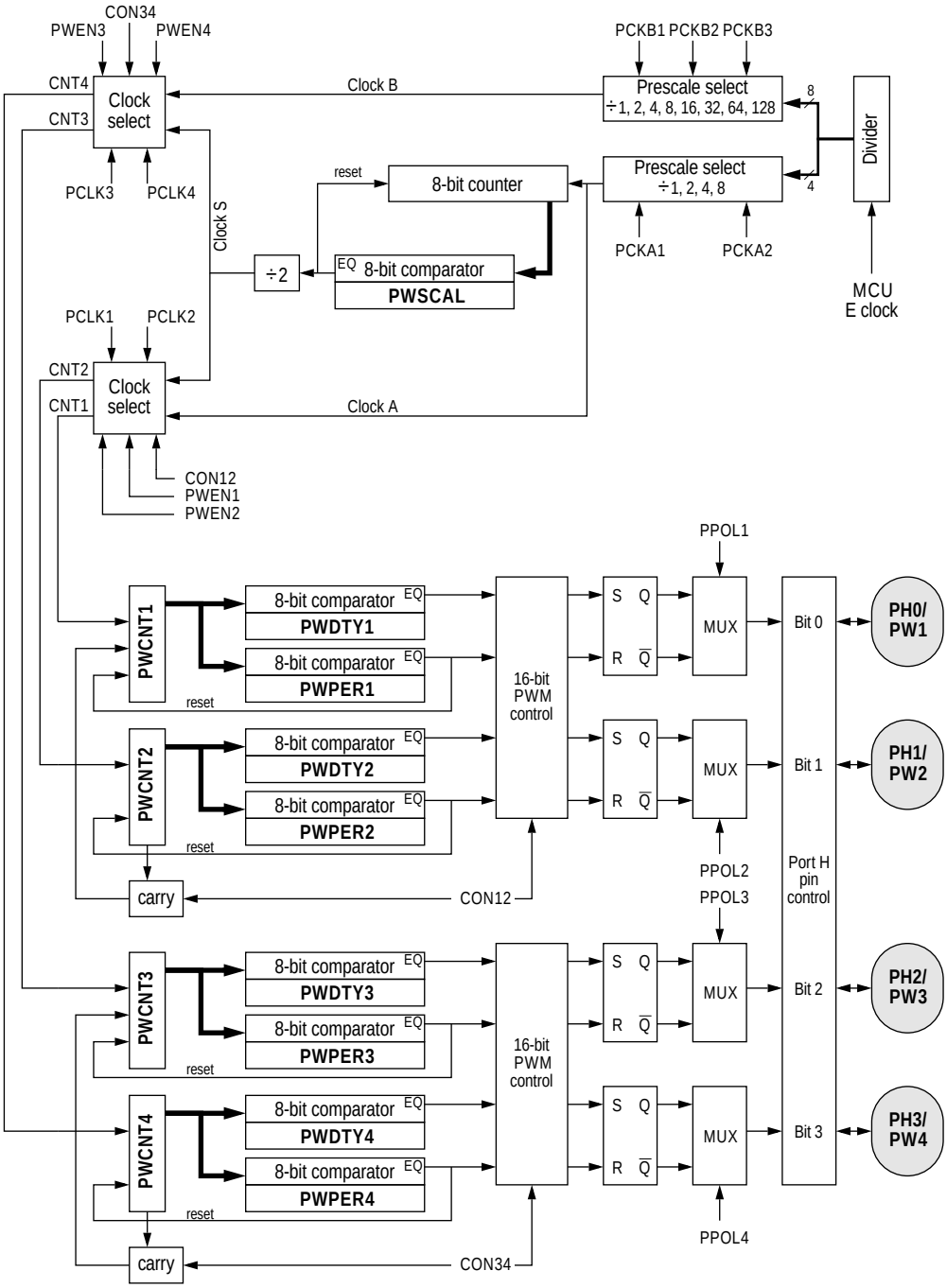


Figure 9-6 PWM timer block diagram

9.7.2.2 Clock prescaler selection

The three available clocks are clock A, clock B, and clock S (scaled). Clock A can be software selected to be E, E/2, E/4, or E/8. Clock B can be software selected to be E, E/2, E/4,..., E/128. The scaled clock (clock S) uses clock A as an input and divides it with a reloadable counter. The rates available are software selectable to be clock A/2, down to clock A /512.

The clock source portion of the block diagram shows the three clock sources and how the scaled clock is created. Clock A is an input to an 8-bit counter which is then compared to a user programmable scale value. When they match, this circuit has an output that is divided by two and the counter is reset.

Each PWM timer channel can be driven by one of two clocks. Refer to Figure 9-6.

PCKA[2:1] — Prescaler for clock A

Determines the frequency of clock A. Refer to Table 9-4.

Bit 3 — Not implemented; always reads zero

PCKB[3:1] — Prescaler for clock B

Determines the frequency of clock B. Refer to Table 9-4.

Table 9-4 Clock A and clock B prescalers

PCKA[2:1]	Clock A
0 0	E
0 1	E/2
1 0	E/4
1 1	E/8

PCKB[3:1]	Clock B
0 0 0	E
0 0 1	E/2
0 1 0	E/4
0 1 1	E/8
1 0 0	E/16
1 0 1	E/32
1 1 0	E/64
1 1 1	E/128

9.7.3 PWPOL — PWM timer polarity & clock source select register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Pulse width polarity select (PWPOL)	$0061	PCLK4	PCLK3	PCLK2	PCLK1	PPOL4	PPOL3	PPOL2	PPOL1	0000 0000

PCLK[4:3] — Pulse width channel 4/3 clock select

1 (set) – Clock S is source.

0 (clear) – Clock B is source.

PCLK[2:1] — Pulse width channel 2/1 clock select

1 (set) – Clock S is source.

0 (clear) – Clock A is source.

PPOL[4:1] — Pulse width channel x polarity

1 (set) – PWM channel x output is high at the beginning of the clock cycle and goes low when duty count is reached.

0 (clear) – PWM channel x output is low at the beginning of the clock cycle and goes high when duty count is reached.

Each channel has a polarity bit that allows a cycle to start with either a high or a low level. This is shown on the block diagram, Figure 9-6, as a selection of either the Q output or the $\overline{Q}$ output of the PWM output flip flop. When one of the bits in the PWPOL register is set, the associated PWM channel output is high at the beginning of the clock cycle, then goes low when the duty count is reached.

9.7.4 PWSCAL — PWM timer prescaler register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Pulse width scale (PWSCAL)	$0062	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	0000 0000

Scaled clock S is generated by dividing clock A by the value in PWSCAL, then dividing the result by two. If PWSCAL = $00, clock A is divided by 256, then divided by two to generate clock S.

9.7.5 PWEN — PWM timer enable register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Pulse width enable (PWEN)	$0063	TPWSL	DISCP	0	0	PWEN4	PWEN3	PWEN2	PWEN1	0000 0000

Each timer has an enable bit to start its waveform output. Writing any of these PWENx bits to one causes the associated port line to become an output regardless of the state of the associated DDR bit. This does not change the state of the DDR bit and when PWENx returns to zero the DDR bit again controls I/O state. On the front end of the PWM timer the clock is connected to the PWM circuit by the PWENx enable bit being high. There is a synchronizing circuit to guarantee that the clock will only be enabled or disabled at an edge.

PWEN contains 4 PWM enable bits — one for each channel. When an enable bit is set to one, the pulse modulated signal becomes available at the associated port pin.

TPWSL — PWM scaled clock test bit (Test mode only)

1 (set) – Clock S output to PWSCAL register (Test only).

0 (clear) – Normal operation.

When TPWSL is one, clock S from the PWM timer is output to PWSCAL register. Normal writing to the PWSCAL register still functions.

DISCP — Disable compare scaled E clock (Test mode only)

1 (set) – Match of period does not reset associated count register (Test only).

0 (clear) – Normal operation.

Bits [5:4] — Not implemented; always read zero

PWEN[4:1] — Pulse width channels 4–1

1 (set) – Channel enabled on the associated port pin.

0 (clear) – Channel disabled.

9.7.6 PWCNT1–4 — PWM timer counter registers 1 to 4

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Pulse width count 1 (PWCNT1)	$0064	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	0000 0000
Pulse width count 2 (PWCNT2)	$0065	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	0000 0000
Pulse width count 3 (PWCNT3)	$0066	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	0000 0000
Pulse width count 4 (PWCNT4)	$0067	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	0000 0000

Each channel has its own counter which can be read at any time without affecting the count or the operation of the PWM channel. Writing to a counter causes it to be reset to $00; this is generally done before the counter is enabled. A counter may also be written to whilst it is enabled; this may cause a truncated PWM period.

9.7.7 PWPER1–4 — PWM timer period registers 1 to 4

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Pulse width period 1 (PWPER1)	$0068	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	1111 1111
Pulse width period 2 (PWPER2)	$0069	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	1111 1111
Pulse width period 3 (PWPER3)	$006A	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	1111 1111
Pulse width period 4 (PWPER4)	$006B	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	1111 1111

There is one period register for each channel. The value in this register determines the period of the associated PWM timer channel. PWPERx is connected internally to a buffer which compares directly with the counter register. The period value in PWPERx is loaded into the buffer when the counter is cleared by the termination of the previous period or by a write to the counter. This register can be written at any time, and the written value will take effect from the start of the next PWM timer cycle. Reads of this register return the most recent value written.

9.7.8 PWDTY1–4 — PWM timer duty cycle registers 1 to 4

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Pulse width duty 1 (PWDTY1)	$006C	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	1111 1111
Pulse width duty 2 (PWDTY2)	$006D	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	1111 1111
Pulse width duty 3 (PWDTY3)	$006E	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	1111 1111
Pulse width duty 4 (PWDTY4)	$006F	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	1111 1111

There is one duty register for each channel. The value in this register determines the duty cycle of the associated PWM timer channel. PWDTYx is compared to the counter contents and if they are equal, a match occurs and the output goes to the state defined by the associated polarity bit. If the register is written while the channel is enabled, then the new value is held in a buffer until the counter rolls over or the channel is disabled. Reads of this register return the most recent value written.

Note: If PWDTYx ≥ PWPERx then there will be no change of state due to the duty cycle value. In addition, if the duty register is set to $00, then the output will always be in the state which would normally be result from the duty change of state (see also Section 9.7.9).

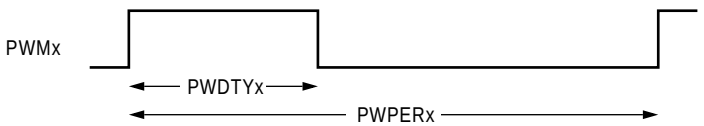


Figure 9-7 PWM duty cycle

9.7.9 Boundary cases

The following boundary conditions apply to the values stored in the PWDTYx and PWPERx registers and the PPOLx bits:

- If PWDTYx = $00, PWPERx > $00 and PPOLx = 0 then the output is always high.
- If PWDTYx = $00, PWPERx > $00 and PPOLx = 1 then the output is always low.
- If PWDTYx ≥ PWPERx and PPOLx = 0 then the output is always low.
- If PWDTYx ≥ PWPERx and PPOLx = 1 then the output is always high.
- If PWPERx = $00 and PPOLx = 0 then the output is always low.
- If PWPERx = $00 and PPOLx = 1 then the output is always high.

10
ANALOG-TO-DIGITAL CONVERTER

The analog-to-digital converter system consists of a single 10-bit successive approximation type converter and a 16-channel multiplexer. Ten of the channels are connected to pins on the MC68HC11KW1 (ports E and G), two are unused and the remaining four channels are dedicated to internal reference points or test functions. The A/D converter shares input pins with port E and port G as follows:

Pin	Alternative function
PG6	AN0
PG7	AN1
PE0	AN2
PE1	AN3
PE2	AN4
PE3	AN5
PE4	AN6
PE5	AN7
PE6	AN8
PE7	AN9

There are eight 10-bit result registers, and control logic allows for four or eight consecutive conversions before stopping, or for conversions to continue with the newest conversion overwriting the oldest result register. Also, conversions can be performed several times on a single selected channel, or consecutively on a selected group of four channels. In addition, the control logic can convert eight channels and then stop, or convert continuously.

Two dedicated lines (VRH and VRL) are provided for the reference voltage inputs. These pins may be connected to a separate or isolated power supply to ensure full accuracy of the A/D conversion. Furthermore, the A/D converter supply pins VDDAD and VSSAD allow the user to isolate the A/D voltage supply from the main VDD/VSS supply lines. This reduces the effects that noise from the CPU core has on the A/D converter, thus improving accuracy.

The 10-bit A/D converter accepts analog inputs ranging from V_{RL} to V_{RH}. Smaller input ranges can also be obtained by adjusting V_{RL} and V_{RH} to the desired upper and lower limits. Conversion is specified and tested for V_{RL} = 0 volts and V_{RH} = 5 volts. The A/D system can be operated with V_{RL} below V_{DD} and/or V_{RL} above V_{SS} as long as V_{RH} is above V_{RL} by enough to support the conversions (2.5 to 5.0 volts).

Each set of four conversions takes 144 cycles of the E-clock, provided that E is greater than or equal to 750 kHz. If E is less than 750 kHz, an internal R-C oscillator, which is nominally 1.5 MHz, must be used for the A/D conversion clock. When the internal R-C oscillator is being used as the conversion clock, the conversion complete flag (CCF) must be used to determine when a conversion sequence has been completed. When using the internal R-C oscillator for A/D conversions, the sample and conversion process runs at the nominal 1.5 MHz rate; however, the conversion results must be transferred to the MCU result registers synchronously with the MCU E-clock, so conversion time is limited to a maximum of one channel per E-clock cycle. Alternatively, if the R-C oscillator is not being used and E is greater than 2.1 MHz, the conversion frequency can be halved to E/2 under control of the ADER bit in the ADFRQ register. Note that in this operating mode, each set of four conversions takes 288 cycles of the E clock.

Two control bits in the OPTION register control the basic configuration of the A/D system. The A/D power-up bit (ADPU) allows the system to be disabled, resulting in reduced power consumption when the A/D system is not being used. Any conversion which is in process when ADPU is written to zero will be aborted. A delay of typically 100 microseconds is required after turning on the A/D (by writing ADPU from 0 to 1) for the analog and comparator sections to stabilize. The CSEL bit is used to select either the internal R-C oscillator or the MCU E-clock as the A/D system clock source.

10.1 Conversion process

The A/D converter is ratiometric. An input voltage equal to V_{RH} converts to $FFC0 (full scale) and an input voltage equal to V_{RL} converts to $0000. An input voltage greater than V_{RH} will convert to $FFC0 with no overflow indication. Note that the six least significant bits always read zero. For ratiometric conversions, the source of each analog input should use V_{RH} as the supply voltage and be referenced to V_{RL}.

The A/D reference inputs are applied to a precision internal digital-to-analog converter. Control logic drives this D/A and the analog output is successively compared to the selected analog input which was sampled at the beginning of the conversion time. The conversion process is monotonic with no missing codes.

10.2 Channel assignments

A multiplexer allows the single A/D converter to select one of sixteen analog signals. Ten of these channels are supported on Port E and G input pins. Of the six other channels, two are reserved for future use and four are for internal reference points and testing purposes. Table 10-1 shows the signals selected by the channel select bits (CD, CC, CB, CA) in the ADCTL register. The CONV8 bit selects either four or eight conversions. All “reserved” channels are connected to V_{RL}.

Table 10-1 Channel assignments

CD	CC	CB	CA	Channel signal	CONV8 = 0 Result in register if		CONV8 = 1 Result in register if	
					MULT = 1	MULT = 0	MULT = 1	MULT = 0
0	0	0	0	AN2 (on PE0)	ADR5	ADR5 – 8	ADR1	ADR1 – 8
0	0	0	1	AN3 (on PE1)	ADR6	ADR5 – 8	ADR2	ADR1 – 8
0	0	1	0	AN4 (on PE2)	ADR7	ADR5 – 8	ADR3	ADR1 – 8
0	0	1	1	AN5 (on PE3)	ADR8	ADR5 – 8	ADR4	ADR1 – 8
0	1	0	0	AN6 (on PE4)	ADR5	ADR5 – 8	ADR5	ADR1 – 8
0	1	0	1	AN7 (on PE5)	ADR6	ADR5 – 8	ADR6	ADR1 – 8
0	1	1	0	AN8 (on PE6)	ADR7	ADR5 – 8	ADR7	ADR1 – 8
0	1	1	1	AN9 (on PE7)	ADR8	ADR5 – 8	ADR8	ADR1 – 8
1	0	0	0	AN0 (on PG6)	ADR5	ADR5 – 8	ADR1	ADR1 – 8
1	0	0	1	AN1 (on PG7)	ADR6	ADR5 – 8	ADR2	ADR1 – 8
1	0	1	0	Reserved	ADR7	ADR5 – 8	ADR3	ADR1 – 8
1	0	1	1	Reserved	ADR8	ADR5 – 8	ADR4	ADR1 – 8
1	1	0	0	VRH(1)	ADR5	ADR5 – 8	ADR5	ADR1 – 8
1	1	0	1	VRL(1)	ADR6	ADR5 – 8	ADR6	ADR1 – 8
1	1	1	0	VRH/2(1)	ADR7	ADR5 – 8	ADR7	ADR1 – 8
1	1	1	1	Test (reserved)(1)	ADR8	ADR5 – 8	ADR8	ADR1 – 8

(1) Used for factory testing.

10.3 Single channel operation

Single channel operation is selected by writing a zero to the MULT bit in the A/D control and status register (ADCTL). This mode has four variations, which can be selected using the CONV8 and SCAN bits in the ADCTL register. In the first two variations, the CONV8 bit is clear and the single selected channel is converted four consecutive times. In the second two variations, the CONV8 bit is set and the single selected channel is converted eight consecutive times. The state of the SCAN bit determines whether continuous or single scanning is selected. The channel is selected by the CD – CA bits in the ADCTL register.

If channels eight and nine are selected, then the result registers previously used for two of the other channels become overwritten with channel eight and nine results.

10.3.1 4-conversion, single scan

Whichever port E or G pin is selected, the first result will be stored in the ADR5 result register and the fourth result will be stored in the ADR8 register. After the fourth conversion is complete all conversion activity is halted until a new conversion command is written to the ADCTL control register.

10.3.2 4-conversion, continuous scan

Conversions continue to be performed on the selected channel with the fifth conversion being stored in the ADR5 register (overwriting the first conversion result), the sixth conversion overwrites ADR6, the seventh overwrites ADR7, and so on continuously. Using this variation, the data in any result register is at most four conversion times old.

10.3.3 8-conversion, single scan

The result of the first conversion will be placed in result register ADR1, while the result of the eighth conversion will be placed in result register ADR8. After the eighth conversion is complete all conversion activity is halted until a new conversion command is written to the ADCTL control register.

10.3.4 8-conversion, continuous scan

Conversions continue to be performed on the selected channel with the ninth conversion being stored in the ADR1 register (overwriting the first conversion result), the tenth conversion overwrites ADR2, the eleventh overwrites ADR3, and so on continuously. Using this variation, the data in any result register is at most eight conversion times old.

10.4 Multiple channel operation

Multiple channel operation is selected by writing a one to the MULT bit in the A/D control and status register (ADCTL). This mode has four variations, which can be selected using the CONV8 and SCAN bits in the ADCTL register. In the first two variations, the CONV8 bit is clear and a group of four channels is selected. In this multiple channel mode, the group of channels selected depends only on the two most significant bits of the channel address (CD and CC). In the second two variations, the CONV8 bit is set and a group of eight channels is selected, depending on the state of the CD bit. The state of the SCAN bit determines whether continuous or single scanning is selected. Refer to Table 10-1.

10.4.1 4-channel single scan

A group of four channels is selected by the CD and CC bits in the ADCTL register (see Table 10-1). The first result is stored in the ADR5 result register, the second in ADR6, the third in ADR7 and the fourth in the ADR8 register. After the fourth conversion is complete, all conversion activity is halted until a new conversion command is written to the ADCTL control register.

10.4.2 4-channel continuous scan

Conversions continue to be performed on the selected group of channels with the fifth conversion being stored in the ADR5 register (replacing the earlier conversion result for the first channel in the group), the sixth conversion overwrites ADR6, the seventh overwrites ADR7, and so on, continuously. Using this second variation the data in any result register is, at most, four conversion times old.

10.4.3 8-channel single scan

When CONV8 is set and MULT is set, then a group of eight channels is converted. The group is selected by the CD bit. Each of the channels is converted and the result is placed in a separate result register. Port E pin 0 uses result register ADR1, Port E pin 1 uses result register ADR2 and so on. Each channel is converted once, then all conversion activity is halted until a new conversion command is written to the ADCTL control register.

10.4.4 8-channel continuous scan

Conversions continue to be performed on the eight channels with the ninth conversion being stored in the ADR1 register (replacing the earlier conversion result for the first channel in the group), the tenth conversion overwrites ADR2, the eleventh overwrites ADR3, and so on, continuously. Using this second variation the data in any result register is, at most, eight conversion times old.

10.5 Power-up and clock select

A/D power up is controlled by the ADPU bit in the OPTION register. When ADPU is cleared, power to the A/D system is removed. When ADPU is set, the A/D system is enabled. A delay of 100 microseconds is required after turning on the A/D converter, to allow the analog bias voltages to stabilize.

Clock select is controlled by the CSEL bit in the OPTION register. When CSEL is cleared, the A/D system uses the system E-clock. When CSEL is set, the A/D system uses an internal R-C clock

source, nominally 1.5 MHz, in which case the R-C internal clock should be selected. A delay of 10 milliseconds is required, after changing CSEL from zero to one, to allow the R-C oscillator to start and internal bias voltages to settle. If the E-clock frequency drops below 750kHz, then the internal R-C oscillator must be used.

When the A/D system is operating with the MCU E-clock, all switching and comparator operations are synchronized with the MCU clock. This allows the comparator results to be sampled at quiet clock times to minimise the effect of internal switching noise. As the internal R-C oscillator is asynchronous with respect to the MCU clock, internal switching noise is more likely to affect the overall accuracy of the A/D results, when using this oscillator, than when using the E-clock.

10.6 Operation in STOP and WAIT modes

If a conversion sequence is still in process when the MC68HC11KW1 enters the STOP or WAIT mode, the conversion of the current channel is suspended. When the MCU resumes normal operation, that channel is re-sampled and the conversion sequence resumes. As the MCU exits the WAIT mode, the A/D circuits are stable and valid results can be obtained on the first conversion. However, in STOP mode the comparator, charge pump and R-C oscillator are turned off. If the MC68HC11KW1 exits the STOP mode with a delay (as is normal), there will automatically be enough time for these circuits to stabilize before the first conversion. If the MC68HC11KW1 exits the STOP mode with no delay (DLY bit in OPTION register equal to zero) and a stable external clock supplied, the user must allow about 100 microseconds for the A/D circuitry to stabilize and to avoid invalid results.

10.7 Registers

10.7.1 ADCTL — A/D control and status register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
A/D control & status register (ADCTL)	$0030	CCF	CONV8	SCAN	MULT	CD	CC	CB	CA	undefined

This register can be read and written at any time. Note that a write to this register will always clear the CCF bit.

CCF — Conversions complete flag

This flag bit is set automatically after an A/D conversion cycle (four or eight conversions, depending on which conversion mode is selected). If a continuous scan mode is selected, the CCF flag will become set after the first time all four (or eight) registers have been updated, and it will remain set until the

ADCTL register is again written. Each time the ADCTL register is written, this bit is automatically cleared, any current conversion is aborted and a new conversion sequence is started.

CONV8 — Convert 8/convert 4 select bit

1 (set) – Convert 8 channels or one channel 8 times, (uses all 8 result registers).

0 (clear) – Convert 4 channels or one channel 4 times (uses 4 result registers).

SCAN — Continuous scan control

1 (set) – Convert continuously.

0 (clear) – Perform selected number of conversions (4 or 8) and stop.

MULT — Multiple channel/single channel control

1 (set) – Convert the channels in the selected group.

0 (clear) – Convert single channel selected.

CD, CC, CB, CA — Channel select bits

When 4-conversion (CONV8 = 0) and multiple channel (MULT=1) modes are selected, the CB and CA bits have no meaning or effect, and the CD and CC bits specify which of four groups of four channels are to be converted. When 8-conversion (CONV8 = 1) and multiple channel (MULT=1) modes are selected, the CC, CB and CA bits have no meaning or effect. Refer to Table 10-1 for a list of the A/D channel assignments.

10.7.2 ADFRQ — A/D converter frequency select register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
A/D frequency select register (ADFRQ)	$0032	0	0	0	0	0	0	0	ADER	0000 0000

This register can be read and written at any time.

Bits [7:1] — Not implemented; always read zero.

ADER — A/D frequency select

1 (set) – E/2 clock is used for A/D conversions.

0 (clear) – E clock is used for A/D conversions.

This bit improves the accuracy of conversion when the MC68HC11KW1 bus frequency is above 2.1MHz.

Note if the CSEK bit in the OPTION register is set, then the internal R-C oscillator (nominally 1.5MHz) is used for the A/D conversion and the ADER bit has no effect on the frequency.

10.7.3 ADR1 — ADR8 A/D result registers

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
A/D result 1 (ADR1) high	$0040	(Bit 15)	(14)	(13)	(12)	(11)	(10)	(9)	(8)	undefined
A/D result 1 (ADR1) low	$0041	(7)	(6)	0	0	0	0	0	0	uu00 0000
A/D result 2 (ADR2) high	$0042	(Bit 15)	(14)	(13)	(12)	(11)	(10)	(9)	(8)	undefined
A/D result 2 (ADR2) low	$0043	(7)	(6)	0	0	0	0	0	0	uu00 0000
A/D result 3 (ADR3) high	$0044	(Bit 15)	(14)	(13)	(12)	(11)	(10)	(9)	(8)	undefined
A/D result 3 (ADR3) low	$0045	(7)	(6)	0	0	0	0	0	0	uu00 0000
A/D result 4 (ADR4) high	$0046	(Bit 15)	(14)	(13)	(12)	(11)	(10)	(9)	(8)	undefined
A/D result 4 (ADR4) low	$0047	(7)	(6)	0	0	0	0	0	0	uu00 0000
A/D result 5 (ADR5) high	$0048	(Bit 15)	(14)	(13)	(12)	(11)	(10)	(9)	(8)	undefined
A/D result 5 (ADR5) low	$0049	(7)	(6)	0	0	0	0	0	0	uu00 0000
A/D result 6 (ADR6) high	$004A	(Bit 15)	(14)	(13)	(12)	(11)	(10)	(9)	(8)	undefined
A/D result 6 (ADR6) low	$004B	(7)	(6)	0	0	0	0	0	0	uu00 0000
A/D result 7 (ADR7) high	$004C	(Bit 15)	(14)	(13)	(12)	(11)	(10)	(9)	(8)	undefined
A/D result 7 (ADR7) low	$004D	(7)	(6)	0	0	0	0	0	0	uu00 0000
A/D result 8 (ADR8) high	$004E	(Bit 15)	(14)	(13)	(12)	(11)	(10)	(9)	(8)	undefined
A/D result 8 (ADR8) low	$004F	(7)	(6)	0	0	0	0	0	0	uu00 0000

The eight 10-bit result registers are read-only; they can be read at any time, but a write will have no meaning or effect. In each result register, the eight high order bits are in one address location and the remaining two low order bits are in bit locations 6 and 7 of the following address. The six unused bits will always read as zeros. This allows a double byte read to be performed without having to adjust the result.

A
ELECTRICAL SPECIFICATIONS

This section contains the electrical specifications and associated timing information for the standard supply voltage (V_{DD} = 5V ± 5%) MC68HC11KW1.

A.1 Maximum ratings

Rating	Symbol	Value	Unit
Supply voltage (1)	V_{DD}	– 0.3 to +7.0	V
Input voltage (1)	V_{IN}	– 0.3 to +7.0	V
Operating temperature range	T_A	T_L to T_H – 40 to +85	°C
Storage temperature range	T_{STG}	– 50 to +150	°C
Current drain per pin (2) – not VDD, VSS, VDDAD, VSSAD, VRH, VRL or ports H, K,	I_D	25	mA

(1) All voltages are with respect to V_{SS}.

(2) Maximum current drain per pin is for one pin at a time, observing maximum power dissipation limits.

Note: This device contains circuitry designed to protect against damage due to high electrostatic voltages or electric fields. However, it is recommended that normal precautions be taken to avoid the application of any voltages higher than those given in the maximum ratings table to this high impedance circuit. For maximum reliability all unused inputs should be tied to either V_{SS} or V_{DD}.

A.2 Thermal characteristics and power considerations

The average chip junction temperature, T_J, in degrees Celsius can be obtained from the following equation:

$$T_J = T_A + (P_D \bullet \theta_{JA}) \quad [1]$$

where:

T_A = Ambient temperature (°C)

θ_{JA} = Package thermal resistance, junction-to-ambient (°C/W)

P_D = Total power dissipation = $P_{INT} + P_{I/O}$ (W)

P_{INT} = Internal chip power = $I_{DD} \bullet V_{DD}$ (W)

$P_{I/O}$ = Power dissipation on input and output pins (user determined)

An approximate relationship between P_D and T_J (if $P_{I/O}$ is neglected) is:

$$P_D = \frac{K}{T_J + 273} \quad [2]$$

Solving equations [1] and [2] for K gives:

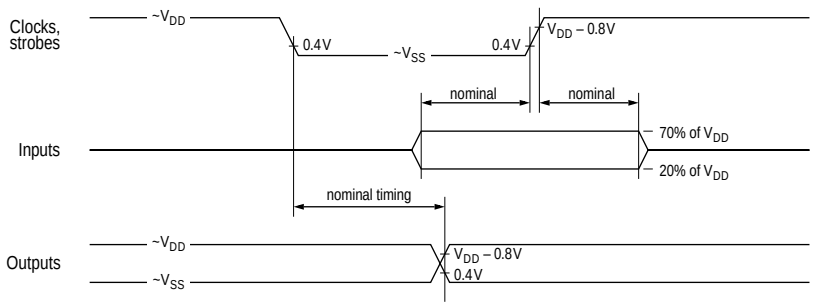
$$K = P_D \bullet (T_A + 273) + \theta_{JA} \bullet P_D^2 \quad [3]$$

where K is a constant for a particular part. K can be determined by measuring P_D (at equilibrium) for a known T_A. Using this value of K, the values of P_D and T_J can be obtained for any value of T_A, by solving the above equations. The package thermal characteristics are shown below:

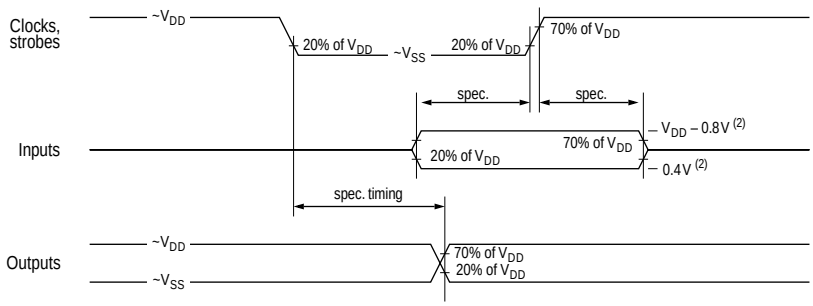
Characteristics	Symbol	Value	Unit
Thermal resistance – 100-pin TQFP package	θ_{JA}	54°	°C/W

A

A.3 Test methods



(b) DC testing



(c) AC testing

Notes:

(1) Full test loads are applied during all DC electrical tests and AC timing measurements.

(2) During AC timing measurements, inputs are driven to 0.4V and V_{DD} – 0.8V; timing measurements are taken at the 20% and 70% of V_{DD} points.

Figure A-1 Test methods

A.4 DC electrical characteristics

(V_{DD} = 5.0Vdc ± 5%, V_{SS} = 0Vdc, $T_A = T_L$ to T_H, unless otherwise noted)

Characteristic	Symbol	Min.	Max.	Unit
Output voltage[1] (I_{LOAD} = ± 10 µA)				
All outputs except XTAL	V_{OL}	—	0.1	V
All outputs except XTAL, $\overline{RESET}$ & MODA	V_{OH}	V_{DD} – 0.1	—	V
Output high voltage[1] (I_{LOAD} = –0.8mA, V_{DD} =4.5V)				
All outputs except XTAL, $\overline{RESET}$ & MODA	V_{OH}	V_{DD} – 0.8	—	V
Output low voltage (I_{LOAD} = +1.6mA)				
All outputs except XTAL	V_{OL}	—	0.4	V
Input high voltage	V_{IH}			V
All inputs except $\overline{RESET}$		0.7V_{DD}	V_{DD} + 0.3	
$\overline{RESET}$		0.8V_{DD}	V_{DD} + 0.3	
Input low voltage – all inputs	V_{IL}	V_{SS} – 0.3	0.2V_{DD}	V
I/O ports three-state leakage ($V_{IN} = V_{IH}$ or V_{IL})	I_{OZ}	—	±10	µA
Input leakage[2]	I_{IN}			
($V_{IN} = V_{DD}$ or V_{SS}) $\overline{IRQ}$, $\overline{XIRQ}$		—	±1	µA
($V_{IN} = V_{DD}$ or V_{SS}) MODB/V_{STBY}		—	±10	µA
Input current with pull-up resistors ($V_{IN} = V_{IL}$)				
Ports B, F, H, Port G[5:0]	I_{IPR}	100	500	µA
RAM stand-by voltage (power down)	V_{SB}	2.0	V_{DD}	V
RAM stand-by current (power down)	I_{SB}	—	20	µA
Input capacitance	C_{in}			
PE[7:0], PG[7,6], $\overline{IRQ}$, $\overline{XIRQ}$, EXTAL		—	8	pF
Ports A, B, C, D, F, G[5:0], H, J, K, MODA/$\overline{LIR}$, $\overline{RESET}$		—	12	pF
Output load capacitance	C_L			pF
All outputs except PD[4:1], XOUT, XTAL, MODA/LIR		—	90	
XOUT		—	30	
PD[4:1]		—	200	
Maximum total supply current[3]:				
RUN:	I_{DD}			
Single chip mode		—	40	mA
Expanded mode		—	50	mA
WAIT: (All peripheral functions shut down)	W_{IDD}			
Single chip mode		—	25	mA
Expanded mode		—	30	mA
STOP:	S_{IDD}			
Single chip mode, no clocks		—	100	µA
Maximum power dissipation	P_D			
Single chip mode		—	210	mW
Expanded mode		—	262.5	mW

(1) V_{OH} specification for $\overline{RESET}$ and MODA is not applicable as they are open-drain pins.

(2) Refer to A/D specification for the leakage current on A/D inputs.

(3) EXTAL is driven with a square wave. No dc loads. Expansion bus active. All other ports configured as inputs; bus frequency is f_0 = 4.0MHz.

A.5 Control timing

(V_{DD} = 5.0V ± 5%, V_{SS} = 0Vdc, $T_A = T_L$ to T_H unless otherwise noted)

Characteristic (1)	Symbol	4.0 MHz		Unit
		Min.	Max.	
Frequency of operation	f_{OP}	0	4.0	MHz
E clock period	t_{CYC}	250	—	ns
Crystal frequency	f_{XTAL}	—	16.0	MHz
External oscillator frequency	$4f_{OP}$	0	16.0	MHz
Processor control setup time (t_{PCSU} = 1/4 t_{CYC} + 50 ns)	t_{PCSU}	112	–	ns
Reset input pulse width (2)	PW_{RSTL} (3) PW_{RSTL} (4)	16 1	— —	t_{CYC}
Mode programming set-up time	t_{MPS}	2	—	t_{CYC}
Mode programming hold time	t_{MPH}	10	—	ns
Interrupt pulse width ($\overline{IRQ}$ edge sensitive mode)	PW_{IRQ}	270	—	ns
Interrupt pulse period(5)	t_{ILIH}	Note 5	—	t_{CYC}
WAIT recovery start-up time	t_{WRS}	—	4	t_{CYC}
Timer pulse width, input capture pulse accumulator input $PW_{TIM} = t_{CYC} + 20$ ns	PW_{TIM}	270	—	ns

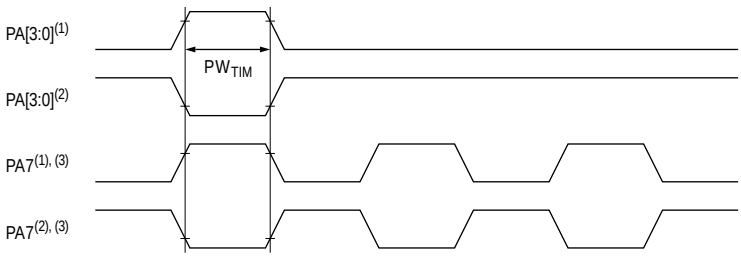
(1) All timing is given with respect to 20% and 70% of V_{DD}, unless otherwise noted.

(2) Reset is recognized during the first clock cycle it is held low. Internal circuitry then drives the pin low for eight clock cycles, releases the pin and samples the pin level four cycles later to determine the source of the interrupt. (See Section 5.)

(3) To guarantee an external reset vector.

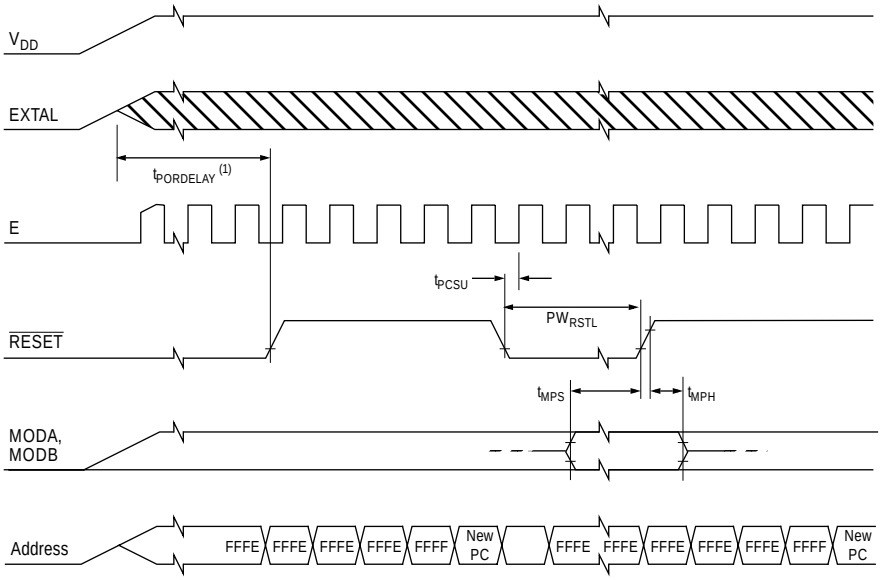
(4) This is the minimum input time; it can be pre-empted by an internal reset.

(5) The minimum period t_{ILIH} should not be less than the number of cycles it takes to execute the interrupt service plus 21 t_{CYC}.



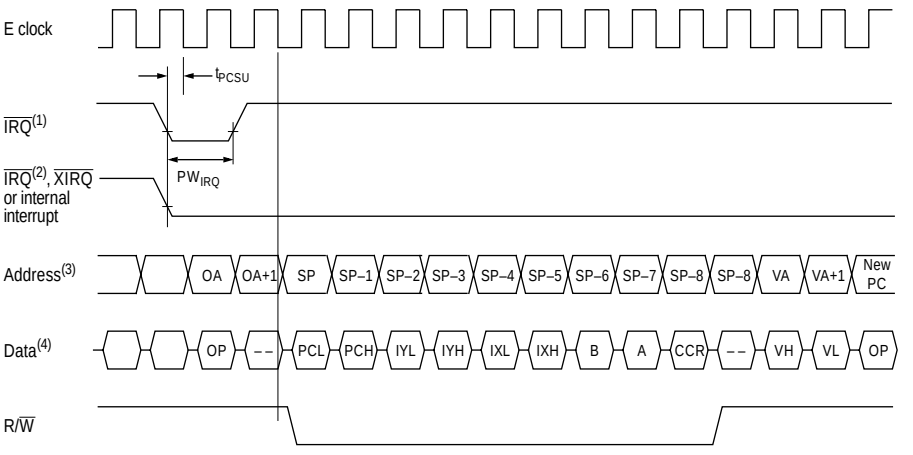
Notes
(1) Rising edge sensitive input.
(2) Falling edge sensitive input.
(3) Maximum pulse accumulator clocking rate is E clock frequency divided by two (E/2).

Figure A-2 Timer inputs



(1) $t_{PORDELAY} = 4064\ t_{CYC}$

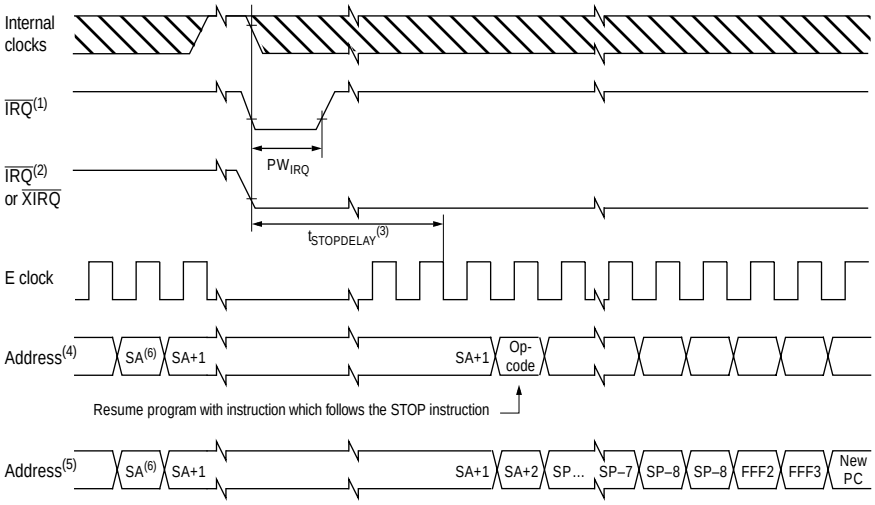
Figure A-3 Reset timing



Notes:
(1) Edge sensitive IRQ pin (IRQE = 1).
(2) Level sensitive IRQ pin (IRQE = 0).
(3) Where OA = Opcode address and VA = Vector address.
(4) Where OP = Opcode, VH = Vector (MSB) and VL = Vector (LSB).

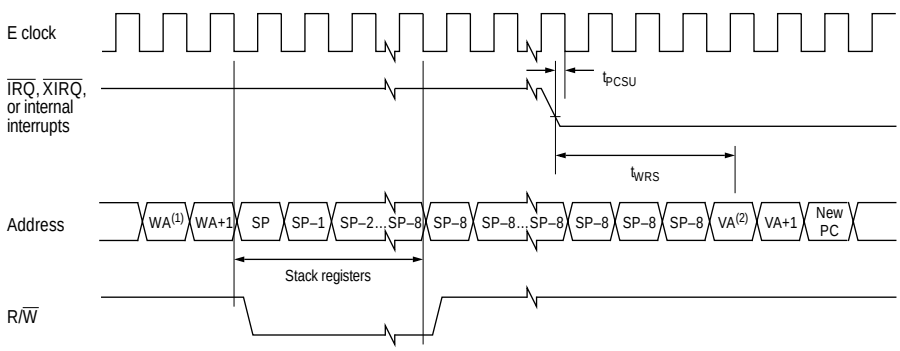
Figure A-4 Interrupt timing

A



Notes:
(1) Edge sensitive $\overline{IRQ}$ pin (IRQE = 1).
(2) Level sensitive $\overline{IRQ}$ pin (IRQE = 0).
(3) If DLY = 1: $t_{STOPDELAY} = 4064\ t_{CYC}$
If DLY = 0: $t_{STOPDELAY} = 4\ t_{CYC}$
(4) $\overline{XIRQ}$ with X-bit in CCR = 1.
(5) $\overline{IRQ}$ (or $\overline{XIRQ}$, with X-bit = 0; in this case vector fetch will be $FFF4/5).
(6) SA = STOP address.

Figure A-5 STOP recovery timing



Notes:
$\overline{RESET}$ also causes recovery from WAIT. (1) WA = WAIT address. (2) VA = Vector address.

Figure A-6 WAIT recovery timing

A.5.1 Peripheral port timing

(V_{DD} = 5.0Vdc ± 5%, V_{SS} = 0Vdc, $T_A = T_L$ to T_H)

Characteristic [1]	Symbol	4.0 MHz		Unit
		Min.	Max.	
Frequency of operation (E clock frequency)	f_{OP}	0	4.0	MHz
E clock period	t_{CYC}	250	—	ns
Peripheral data set-up time, all ports [2]	t_{PDSU}	100	—	ns
Peripheral data hold time, all ports [2]	t_{PDH}	50	—	ns
Delay time, peripheral data write	t_{PWD}			
MCU write to port A, B, G[5:0], H, J or K		—	200	ns
MCU write to port C, D or F ($t_{PWD} = t_{CYC}/4 + 100ns$)		—	162	ns

(1) All timing is given with respect to 20% and 70% of V_{DD}, unless otherwise noted.

(2) Port C, and D timing is valid for active drive (CWOM, DWOM, and WOMS bits clear).

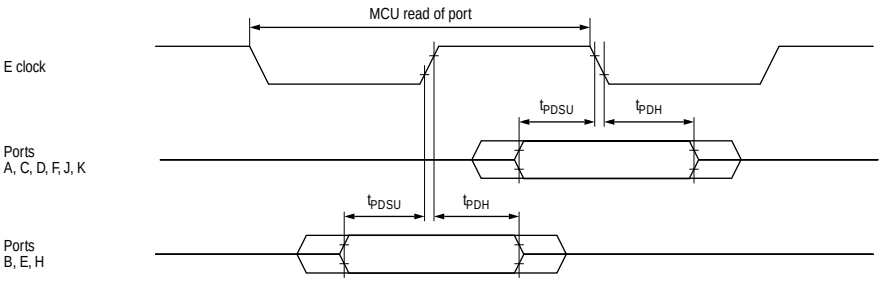


Figure A-7 Port read timing diagram

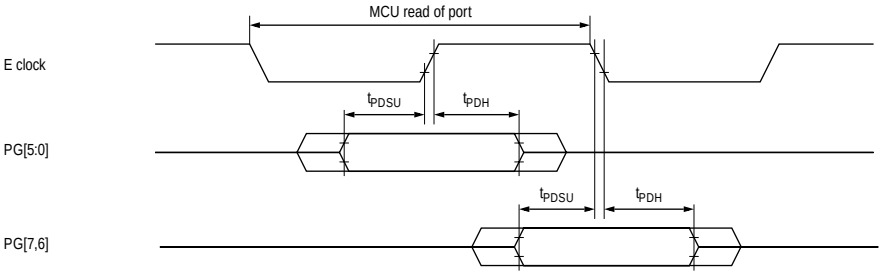


Figure A-8 Port G control timing

A

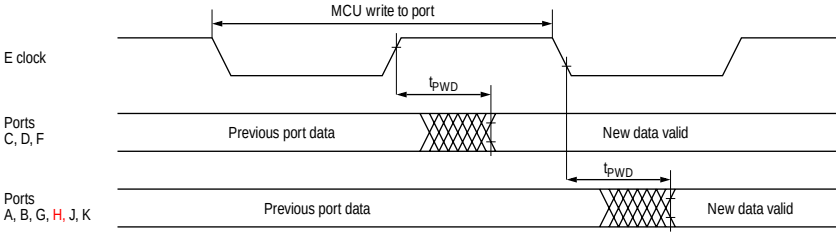


Figure A-9 Port write timing diagram

A

A.5.2 Analog-to-digital converter characteristics

(V_{DD} = 5.0V ± 5%, V_{SS} = 0V, V_{RH} = V_{DD}, V_{RL} = V_{SS}, T_A = T_L to T_H, 750kHz ≤ E ≤ 4MHz, unless otherwise noted(1))

Characteristic	Parameter	Min.	Absolute	Max.	Unit
Resolution	Number of bits resolved by ADC	—	10	—	bits
Non-linearity(2)	Maximum deviation from the best fitting ADC transfer characteristics	—	—	± 1.0	LSB
Offset error(2)	Maximum offset from ADC transfer characteristics	—	—	± 1.0	LSB
Total unadjusted error	Maximum deviation from the ideal ADC transfer characteristics	—	—	± 1.5	LSB
Quantization error	Uncertainty due to converter resolution	—	—	± 0.5	LSB
Absolute accuracy	Difference between the actual input voltage and the full-scale weighted equivalent of the binary output code, including all error sources	—	—	± 2.0	LSB
Conversion range	Analog input voltage range	V_{RL}	—	V_{RH}	V
Analog input voltage(3)	Maximum and minimum analog input voltage	V_{RL} – 0.3	—	1.125V_{RH}	V
V_{RH}	Maximum analog reference voltage	V_{RL}	—	V_{DD} + 0.1	V
V_{RL}	Minimum analog reference voltage	V_{SS} – 0.1	—	V_{RH}	V
ΔV_R	Minimum difference between V_{RH} and V_{RL}	4.5	—	—	V
Monotinicity	Conversion result never decreases with an increase in input voltage and has no missing codes		Guaranteed		
Zero input reading	Conversion result when V_{in} = V_{RL}	0000	—	—	Hex
Full scale reading	Conversion result when V_{in} = V_{RH}	—	—	FFC0	Hex
Input leakage(4)	Input leakage on A/D pins: PE[7:0], PG[7,6]	—	—	400	nA
Input leakage V_{rh} to V_{rl}	V_{rh} to V_{rl} divider chain current consumption			300	µA

(1) In order to attain these accuracies with the ECK > 2.1MHz, the ADER bit in ADFRQ must be set. Input clock duty cycles other than 50% will affect the A/D accuracies

(2) Non-linearity and offset error are characterised using the process window parameters affecting the ADC accuracy, but they are not tested

(3) Minimum analogue input voltage should not go below V_{RL}-0.3V
Maximum analogue input voltage should not exceed 1.125V_{rh}

(4) Source impedance should equal approximately 1 KΩ. Source impedances greater than 1 KΩ affect accuracy adversely because of input leakage.

A

A.5.3 Serial peripheral interface timing

(V_{DD} = 5.0Vdc ± 5%, V_{SS} = 0Vdc, T_A = T_L to T_H)

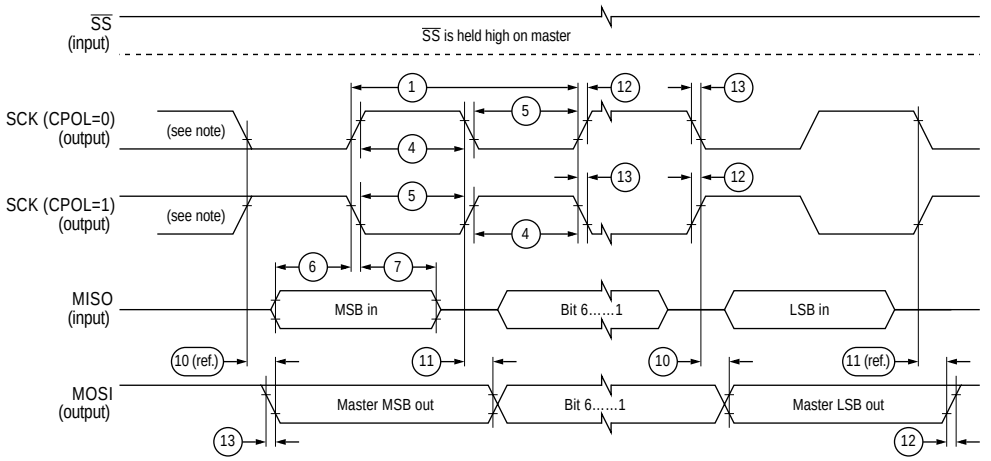
Num	Characteristic (1)		Symbol	4.0MHz		Unit
				Min.	Max.	
	Operating frequency	Master	$f_{OP(M)}$	0	0.5	f_{OP}
		Slave	$f_{OP(S)}$	0	4.0	MHz
1	Cycle time	Master	$t_{CYC(M)}$	2.0	—	t_{CYC}
		Slave	$t_{CYC(S)}$	250	—	ns
2	Enable lead time (2)	Master	$t_{LEAD(M)}$	—	—	ns
		Slave	$t_{LEAD(S)}$	200	—	ns
3	Enable lag time (2)	Master	$t_{LAG(M)}$	—	—	ns
		Slave	$t_{LAG(S)}$	200	—	ns
4	Clock (SCK) high time	Master	$t_{W(SCKH)M}$	130	—	ns
		Slave	$t_{W(SCKH)S}$	85	—	ns
5	Clock (SCK) low time	Master	$t_{W(SCKL)M}$	130	—	ns
		Slave	$t_{W(SCKL)S}$	85	—	ns
6	Input data set-up time	Master	$t_{SU(M)}$	100	—	ns
		Slave	$t_{SU(S)}$	100	—	ns
7	Input data hold time	Master	$t_{H(M)}$	100	—	ns
		Slave	$t_{H(S)}$	100	—	ns
8	Access time (from high-z to data active)	Slave	t_A	0	120	ns
9	Disable time (hold time to high-z state)	Slave	t_{DIS}	—	125	ns
10	Data valid (after enable edge) (3)		$t_{V(S)}$	—	125	ns
11	Output data hold time (after enable edge)		t_{HO}	0	—	ns
12	Rise time (3)					
	SPI outputs (SCK, MOSI and MISO)		t_{RM}	—	100	ns
	SPI inputs (SCK, MOSI, MISO and $\overline{SS}$)		t_{RS}	—	2.0	µs
13	Fall time (3)					
	SPI outputs (SCK, MOSI and MISO)		t_{FM}	—	100	ns
	SPI inputs (SCK, MOSI, MISO and $\overline{SS}$)		t_{FS}	—	2.0	µs

(1) All timing is given with respect to 20% and 70% of V_{DD}, unless otherwise noted.

(2) Signal production depends on software.

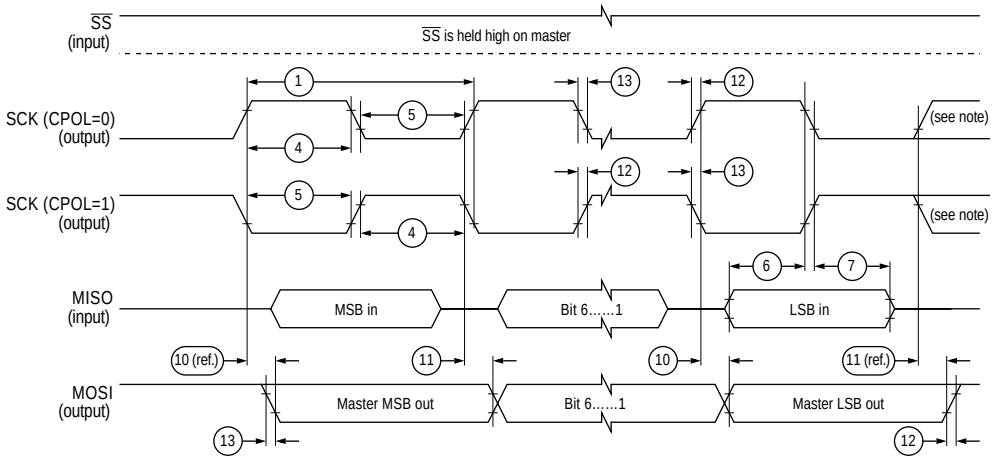
(3) Assumes 200 pF load on all SPI pins.

A



Note: This first clock edge is generated internally, but is not seen at the SCK pin.

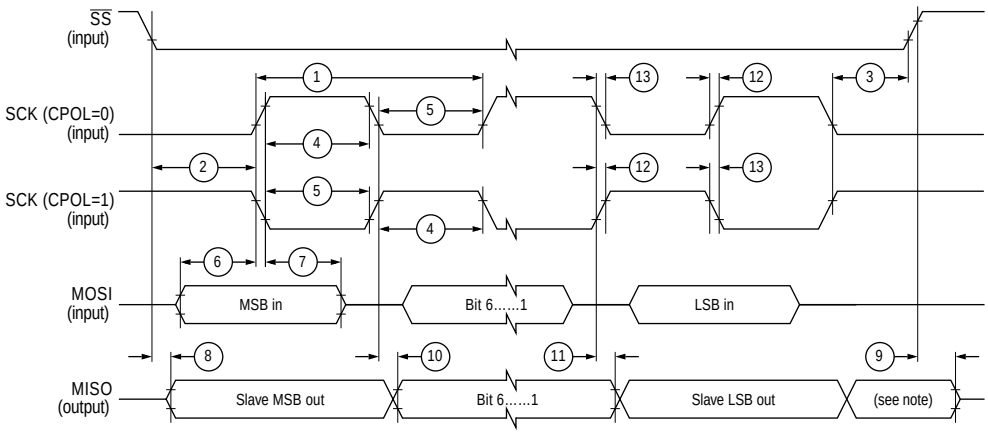
Figure A-10 SPI master timing (CPHA = 0)



Note: This last clock edge is generated internally, but is not seen at the SCK pin.

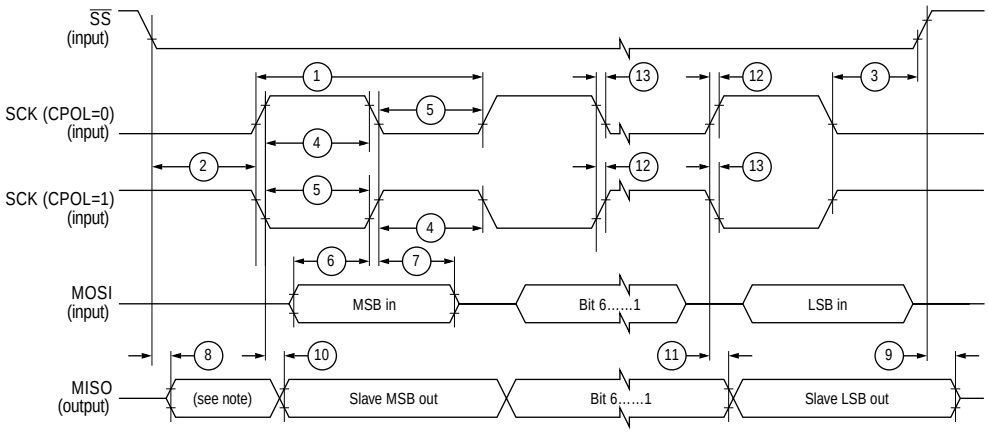
Figure A-11 SPI master timing (CPHA = 1)

A



Note: Not defined, but normally the MSB of character just received.

Figure A-12 SPI slave timing (CPHA = 0)



Note: Not defined, but normally the LSB of character last transmitted.

Figure A-13 SPI slave timing (CPHA = 1)

A

A.5.4 Non-multiplexed expansion bus timing

(V_{DD} = 5.0Vdc ± 5%, V_{SS} = 0Vdc, $T_A = T_L$ to T_H)

Num	Characteristic [1]	Symbol	4.0MHz		Unit
			Min.	Max.	
	Frequency of operation (E clock frequency)	f_{OP}	0	4.0	MHz
1	E clock period	t_{CYC}	250	—	ns
2	Pulse width, E low [2], [3]	PW_{EL}	105	—	ns
3	Pulse width, E high [2], [3], [4]	PW_{EH}	100	—	ns
4A	E clock rise time	t_R	—	20	ns
4B	E clock fall time	t_F	—	15	ns
9	Address hold time [3]	t_{AH}	21	—	ns
11	Address delay time [3]	t_{AD}	—	71	ns
12	Address valid to E rise time [3]	t_{AV}	34	—	ns
17	Read data set-up time	t_{DSR}	20	—	ns
18	Read data hold time	t_{DHR}	0	—	ns
19	Write data delay time	t_{DDW}	—	40	ns
21	Write data hold time [3]	t_{DHW}	31	—	ns
29	MPU address access time [3], [4]	t_{ACCA}	144	—	ns
39	Write data set-up time [3], [4]	t_{DSW}	60	—	ns
50	E valid chip select delay time	t_{ECSD}	—	45	ns
51	E valid chip select access time [4]	t_{ECSA}	40	—	ns
52	Chip select hold time	t_{CH}	0	25	ns
54	Address valid chip select delay time	t_{ACSD}	—	103	ns
55	Address valid chip select access time [4]	t_{ACSA}	113	—	ns
56	Address valid to chip select time	t_{AVCS}	10	—	ns
57	Address valid to data three-state time	t_{AVDZ}	—	10	ns

(1) All timing is given with respect to 20% and 70% of V_{DD}, unless otherwise noted.

(2) Input clock duty cycles other than 50% will affect the bus performance.

(3) For $f_{OP} \leq 2$MHz the following formulae may be used to calculate parameter values:

$PW_{EL} = t_{CYC}/2 - 20$ns — $PW_{EH} = t_{CYC}/2 - 25$ns

$t_{AH} = t_{CYC}/8 - 10$ns — $t_{AD} = t_{CYC}/8 + 40$ns

$t_{AV} = PW_{EL} - t_{AD}$ — $t_{DHW} = t_{CYC}/8$

$t_{ACCA} = t_{CYC} - t_F - t_{DSR} - t_{AD}$ — $t_{DSW} = PW_{EH} - t_{DDW}$

$t_{ECSA} = PW_{EH} - t_{ECSD} - t_{DSR}$ — $t_{ACSA} = t_{CYC} - t_F - t_{DSR} - t_{ACSD}$

(4) Indicates a parameter affected by clock stretching. Add n(t_{CYC}) to the parameter value, where n = 1, 2 or 3, depending on the valued written to the CSCSTR register.

A

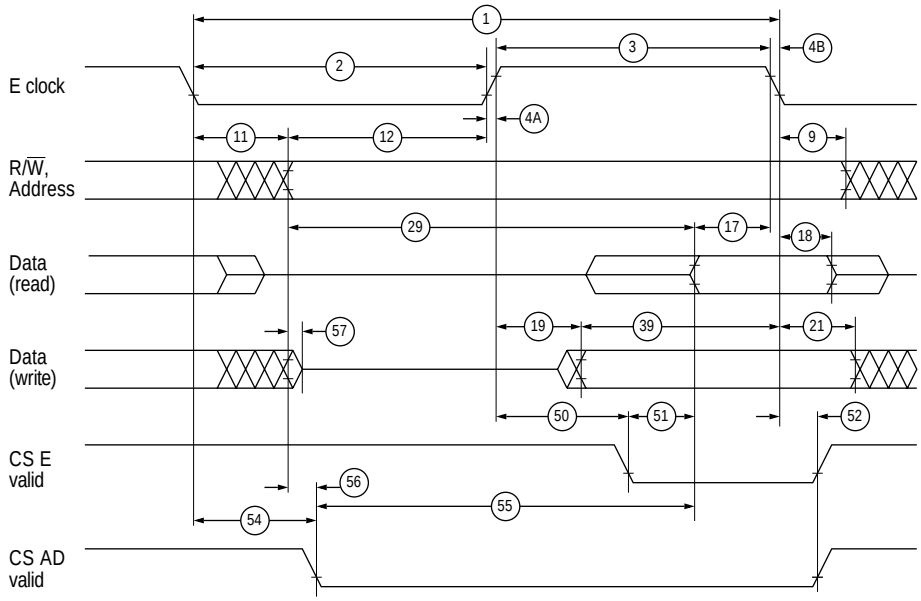


Figure A-14 Expansion bus timing

A.6 EEPROM characteristics

Characteristic	Temperature range −40 to +85°C	Unit
Programming time, t_{EEPROG} (1)		
<1MHz, RCO enabled	10	ms
1–2MHz, RCO disabled	20	
≥2MHz & whenever RCO enabled	10	
Erase time: byte, row and bulk (1)	10	ms
Write/erase endurance (2)	10000	cycles
Data retention (2)	10	years

(1) The RC oscillator (RCO) must be enabled (by setting the CSEL bit in the OPTION register) for EEPROM programming and erasure when the E clock frequency is less than 1.0MHz.

(2) Refer to the current issue of Motorola's quarterly ***Reliability Monitor Report*** for the latest failure rate information.

THIS PAGE LEFT BLANK INTENTIONALLY

B
MECHANICAL DATA

B.1 Packaging

The MC68HC11KW1 is available packaged in a 100-pin thin quad flat pack (TQFP).

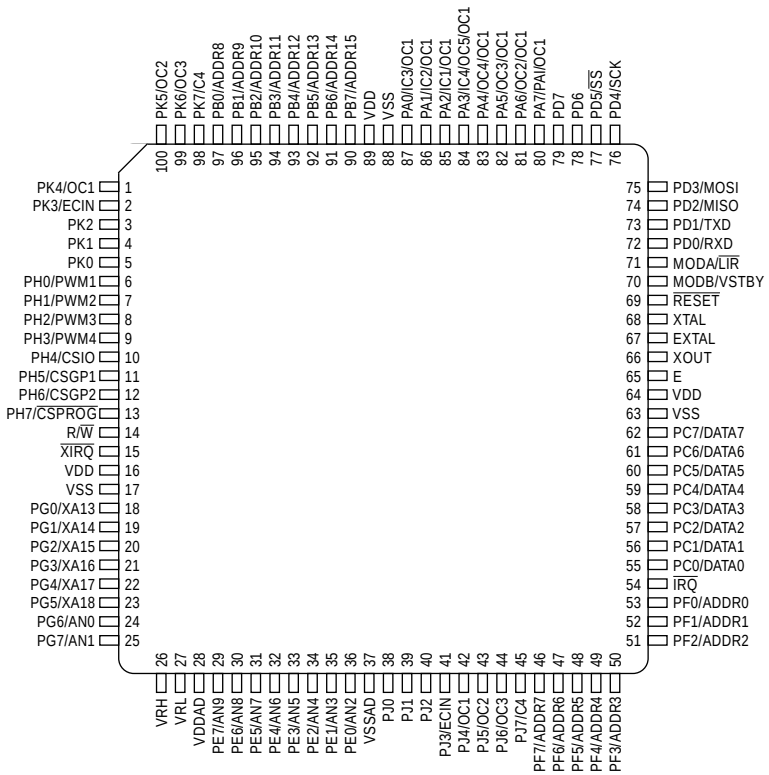
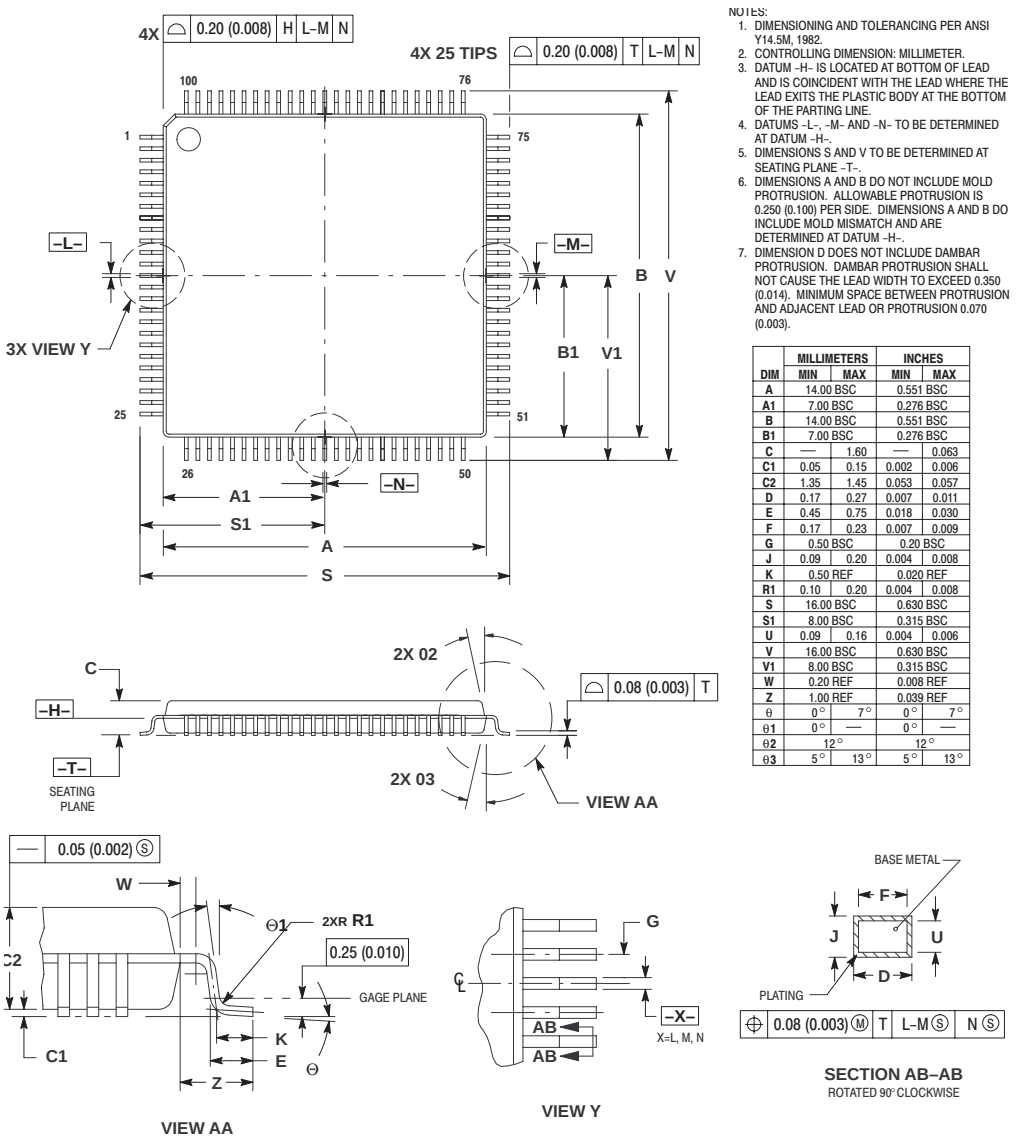


Figure B-1 100-pin TQFP



NOTES:
1. DIMENSIONING AND TOLERANCING PER ANSI Y14.5M, 1982.
2. CONTROLLING DIMENSION: MILLIMETER.
3. DATUM –H– IS LOCATED AT BOTTOM OF LEAD AND IS COINCIDENT WITH THE LEAD WHERE THE LEAD EXITS THE PLASTIC BODY AT THE BOTTOM OF THE PARTING LINE.
4. DATUMS –L–, –M– AND –N– TO BE DETERMINED AT DATUM –H–.
5. DIMENSIONS S AND V TO BE DETERMINED AT SEATING PLANE –T–.
6. DIMENSIONS A AND B DO NOT INCLUDE MOLD PROTRUSION. ALLOWABLE PROTRUSION IS 0.250 (0.100) PER SIDE. DIMENSIONS A AND B DO INCLUDE MOLD MISMATCH AND ARE DETERMINED AT DATUM –H–.
7. DIMENSION D DOES NOT INCLUDE DAMBAR PROTRUSION. DAMBAR PROTRUSION SHALL NOT CAUSE THE LEAD WIDTH TO EXCEED 0.350 (0.014). MINIMUM SPACE BETWEEN PROTRUSION AND ADJACENT LEAD OR PROTRUSION 0.070 (0.003).

	MILLIMETERS		INCHES	
DIM	MIN	MAX	MIN	MAX
A	14.00 BSC		0.551 BSC	
A1	7.00 BSC		0.276 BSC	
B	14.00 BSC		0.551 BSC	
B1	7.00 BSC		0.276 BSC	
C	—	1.60	—	0.063
C1	0.05	0.15	0.002	0.006
C2	1.35	1.45	0.053	0.057
D	0.17	0.27	0.007	0.011
E	0.45	0.75	0.018	0.030
F	0.17	0.23	0.007	0.009
G	0.50 BSC		0.20 BSC	
J	0.09	0.20	0.004	0.008
K	0.50 REF		0.020 REF	
R1	0.10	0.20	0.004	0.008
S	16.00 BSC		0.630 BSC	
S1	8.00 BSC		0.315 BSC	
U	0.09	0.16	0.004	0.006
V	16.00 BSC		0.630 BSC	
V1	8.00 BSC		0.315 BSC	
W	0.20 REF		0.008 REF	
Z	1.00 REF		0.039 REF	
θ	0°	7°	0°	7°
θ1	0°	—	0°	—
θ2	12°		12°	
θ3	5°	13°	5°	13°

Case 983-01

Figure B-2 100-pin TQFP mechanical dimensions

C
DEVELOPMENT SYSTEMS

The following information provides a reference to development tools for the M68HC11 family of microcontrollers. For more detailed information please refer to the appropriate system manual.

Table C-1 M68HC11 development tools

Devices	Evaluation boards	Evaluation modules	Evaluation systems/kits	Programmer boards
MC68HC11KW1	—	M68EM11KW1	—	M68SPGMR11

Note: Target cables for the evaluation module should be ordered separately.

C.1 EVS — Evaluation system

The EVS is an economical tool for designing, debugging and evaluating target systems based on the MC68HC11KW1 device. The two printed circuit boards that comprise the EVS are the M68EM11KW1 emulator module and the M68PFB11KIT platform board. The main features of the EVS are as follows:

- Monitor/debugger firmware
- Single-line assembler/disassembler
- Host computer download capability
- Dual memory maps:
  - 64K byte monitor map that includes 16K bytes of monitor EPROM
- MCU extension I/O port for single chip, expanded and special test operating modes
- RS-232C terminal and host I/O ports
- Logic analyser connector

C.2 MMDS11 — Motorola modular development system

The MMDS11 is an emulator system that provides a bus state analyser and real-time memory windows. The unit's integrated design environment includes an editor, an assembler, user interface and source-level debug. A complete MMDS11 consists of:

- A station module — the metal MMDS11 enclosure, containing the control board and the internal power supply. Most system cables connect to the MMDS11 station module. (The cable to an optional target system, however, runs through an aperture in the station module enclosure to connect directly to the emulator module).
- An emulator module (EM) — such as the EM11KW1: a printed circuit board that enables system functionality for a specific set of MCUs. The EM fits into the station module through a sliding panel in the enclosure top. The EM has a connector for the target cable.
- Two logic clip cable assemblies — twisted pair cables that connect the station module to your target system, a test fixture, a clock or any other circuitry useful for evaluation or analysis. One end of each cable assembly has a moulded connector, which fits into station module pod A or pod B. Leads at the other end of the cable terminate in female probe tips. Ball clips come with the cables.
- A 9-lead RS-232 serial cable — the cable that connects the station module to the host computer's RS-232 port.

C.3 SPGMR11 — Serial peripheral system

The SPGMR11 is an economical tool for programming M68HC11 MCUs. The system consists of the M68SPGMR11 unit and a programming module which adapts the SPGMR11 to the appropriate MCU and package type. For the MC68HC11KW1, the programming module can be ordered as M68PA11KW1PU100 for the 100-pin packaged device.

GLOSSARY

This section contains abbreviations and specialist words used in this data sheet and throughout the industry. Further information on many of the terms may be gleaned from Motorola's ***M68HC11 Reference Manual, M68HC11RM/AD***, or from a variety of standard electronics text books.

$xxxx	The digits following the '$' are in hexadecimal format.
%xxxx	The digits following the '%' are in binary format.
A/D, **ADC**	Analog-to-digital (converter).
Bootstrap mode	In this mode the device automatically loads its internal memory from an external source on reset and then allows this program to be executed.
Byte	Eight bits.
CCR	Condition codes register; an integral part of the CPU.
CERQUAD	A ceramic package type, principally used for EPROM and high temperature devices.
Clear	'0' — the logic zero state; the opposite of 'set'.
CMOS	Complementary metal oxide semiconductor. A semiconductor technology chosen for its low power consumption and good noise immunity.
COP	Computer operating properly. *aka* 'watchdog'. This circuit is used to detect device runaway and provide a means for restoring correct operation.
CPU	Central processing unit.
D/A, DAC	Digital-to-analog (converter).
EEPROM	Electrically erasable programmable read only memory. *aka* 'EEROM'.
EPROM	Erasable programmable read only memory. This type of memory requires exposure to ultra-violet wavelengths in order to erase previous data. *aka* 'PROM'.
ESD	Electrostatic discharge.
Expanded mode	In this mode the internal address and data bus lines are connected to external pins. This enables the device to be used in much more complex systems, where there is a need for external memory for example.

EVS	Evaluation system. One of the range of platforms provided by Motorola for evaluation and emulation of their devices.
HCMOS	High-density complementary metal oxide semiconductor. A semiconductor technology chosen for its low power consumption and good noise immunity.
I/O	Input/output; used to describe a bidirectional pin or function.
Input capture	(IC) This is a function provided by the timing system, whereby an external event is 'captured' by storing the value of a counter at the instant the event is detected.
Interrupt	This refers to an asynchronous external event and the handling of it by the MCU. The external event is detected by the MCU and causes a predetermined action to occur.
$\overline{\text{IRQ}}$	Interrupt request. The overline indicates that this is an active-low signal format.
K byte	A kilo-byte (of memory); 1024 bytes.
LCD	Liquid crystal display.
LSB	Least significant byte.
M68HC11	Motorola's family of advanced 8-bit MCUs.
MCU	Microcontroller unit.
MSB	Most significant byte.
Nibble	Half a byte; four bits.
NRZ	Non-return to zero.
Opcode	The opcode is a byte which identifies the particular instruction and operating mode to the CPU. See also: prebyte, operand.
Operand	The operand is a byte containing information the CPU needs to execute a particular instruction. There may be from 0 to 3 operands associated with an opcode. See also: opcode, prebyte.
Output compare	(OC) This is a function provided by the timing system, whereby an external event is generated when an internal counter value matches a predefined value.
PLCC	Plastic leaded chip carrier package.
PLL	Phase-locked loop circuit. This provides a method of frequency multiplication, to enable the use of a low frequency crystal in a high frequency circuit.
Prebyte	This byte is sometimes required to qualify an opcode, in order to fully specify a particular instruction. See also: opcode, operand.
Pull-down, pull-up	These terms refer to resistors, sometimes internal to the device, which are permanently connected to either ground or V_{DD}.

PWM	Pulse width modulation. This term is used to describe a technique where the width of the high and low periods of a waveform is varied, usually to enable a representation of an analog value.
QFP	Quad flat pack package.
RAM	Random access memory. Fast read and write, but contents are lost when the power is removed.
RFI	Radio frequency interference.
RTI	Real-time interrupt.
ROM	Read-only memory. This type of memory is programmed during device manufacture and cannot subsequently be altered.
RS-232C	A standard serial communications protocol.
SAR	Successive approximation register.
SCI	Serial communications interface.
Set	'1' — the logic one state; the opposite of 'clear'.
Silicon glen	An area in the central belt of Scotland, so called because of the concentration of semiconductor manufacturers and users found there.
Single chip mode	In this mode the device functions as a self contained unit, requiring only I/O devices to complete a system.
SPI	Serial peripheral interface.
Test mode	This mode is intended for factory testing.
TTL	Transistor-transistor logic.
UART	Universal asynchronous receiver transmitter.
VCO	Voltage controlled oscillator.
Watchdog	*see* 'COP'.
Wired-OR	A means of connecting outputs together such that the resulting composite output state is the logical OR of the state of the individual outputs.
Word	Two bytes; 16 bits.
$\overline{\text{XIRQ}}$	Non-maskable interrupt request. The overline indicates that this has an active-low signal format.

THIS PAGE LEFT BLANK INTENTIONALLY

INDEX

In this index numeric entries are placed first; page references in *italics* indicate that the reference is to a figure.

D

E

F

G

H

I

J

L

M

N

O

P

R

S

T

U

V

W

X

Z

Home Page:
www.freescale.com
email:
support@freescale.com
USA/Europe or Locations Not Listed:
Freescale Semiconductor
Technical Information Center, CH370
1300 N. Alma School Road
Chandler, Arizona 85224
(800) 521-6274
480-768-2130
support@freescale.com
Europe, Middle East, and Africa:
Freescale Halbleiter Deutschland GmbH
Technical Information Center
Schatzbogen 7
81829 Muenchen, Germany
+44 1296 380 456 (English)
+46 8 52200080 (English)
+49 89 92103 559 (German)
+33 1 69 35 48 48 (French)
support@freescale.com
Japan:
Freescale Semiconductor Japan Ltd.
Headquarters
ARCO Tower 15F
1-8-1, Shimo-Meguro, Meguro-ku
Tokyo 153-0064, Japan
0120 191014
+81 2666 8080
support.japan@freescale.com
Asia/Pacific:
Freescale Semiconductor Hong Kong Ltd.
Technical Information Center
2 Dai King Street
Tai Po Industrial Estate,
Tai Po, N.T., Hong Kong
+800 2666 8080
support.asia@freescale.com
For Literature Requests Only:
Freescale Semiconductor
Literature Distribution Center
P.O. Box 5405
Denver, Colorado 80217
(800) 441-2447
303-675-2140
Fax: 303-675-2150
LDCForFreescaleSemiconductor
@hibbertgroup.com

RoHS-compliant and/or Pb- free versions of Freescale products have the functionality and electrical characteristics of their non-RoHS-compliant and/or non-Pb- free counterparts. For further information, see http://www.freescale.com or contact your Freescale sales representative.

For information on Freescale.s Environmental Products program, go to http://www.freescale.com/epp.

freescale™
semiconductor