MC68HC11G5/D

MC68HC11G5

HC11

MC68HC11G5
MC68HC11G7
MC68HC711G5

TECHNICAL DATA

TECHNICAL
DATA

MOTOROLA

MOTOROLA

MC68HC11G5
MC68HC11G7
MC68HC711G5

High-density Complementary Metal Oxide Semiconductor (HCMOS) Microcontroller Unit

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Continued)

Section 5
Resets, Interrupts and Low Power Modes

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Continued)

Section 12
CPU, Addressing Modes and Instruction Set

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Concluded)

LIST OF ILLUSTRATIONS

LIST OF ILLUSTRATIONS (Concluded)

LIST OF TABLES

This page intentionally left blank

SECTION 1

INTRODUCTION

1.1 GENERAL

The MC68HC11G5, which includes 16 kilobytes of on-board ROM, is a member of the M68HC11 family of advanced capability MCUs. This device has 512 bytes of on-board static RAM and a non-multiplexed expanded bus for accessing external memory.

The original MC68HC11 timer system is expanded to contain 3 input capture ports, 4 output compare ports and 3 software selectable input capture or output compare ports. A new 4-channel pulse width modulation timer has been added and there is also an event counter timer system.

The serial I/O system consists of an asynchronous NRZ Serial Communications Interface (SCI) plus a synchronous Serial Peripheral Interface (SPI). The A/D converter is enhanced to 10 bits. The MC68HC11G5 also includes a real time interrupt circuit and a Computer Operating Properly (COP) watchdog system, as on the original MC68HC11A8.

The MC68HC11G7, which is similar to the MC68HC11G5 but with 24 kilobytes of ROM instead of 16 kilobytes, is described in Appendix A.

The MC68HC711G5, again similar to the MC68HC11G5 but with 16 kilobytes of EPROM instead of ROM, is described in Appendix B.

This section outlines the general characteristics and special features of the MC68HC11G5 high-density complementary metal oxide semiconductor (HCMOS) microcontroller unit (MCU).

1.2 FEATURES OF THE MOTOROLA MC68HC11G5 MCU

- Low-power/high-performance M68HC11 CPU core
- 16 kilobytes of ROM
- 256 bytes of bootstrap ROM
- 512 bytes of static RAM
- RAM, registers and I/O mappable to any 4k boundary

- Expanded non-multiplexed data/address buses give access to a total address space of 64 kilobytes
- Maximum of seven 8-bit, one 6-bit and one 4-bit I/O ports
- 2 separate 16-bit timers, each with its own E-clock divider circuitry
- External clock option on Timer 2
- 4 output compare (OC), 3 input capture (IC) and 3 selectable OC/IC channels
- 4 PWM timers, each with its own counter and programmable period and duty cycle
- Dual event counter featuring phase-shifter, pulse accumulator, periodic timer and 256 clock divider
- 8-channel, 10-bit A/D converter with 1, 4 and 8-channel scanning modes
- COP watchdog timer and clock monitor
- Serial I/O system incorporating asynchronous NRZ and synchronous SPI
- Single Chip, Expanded, Bootstrap and Test modes of operation
- 84-pin PLCC package (quad surface mount plastic)

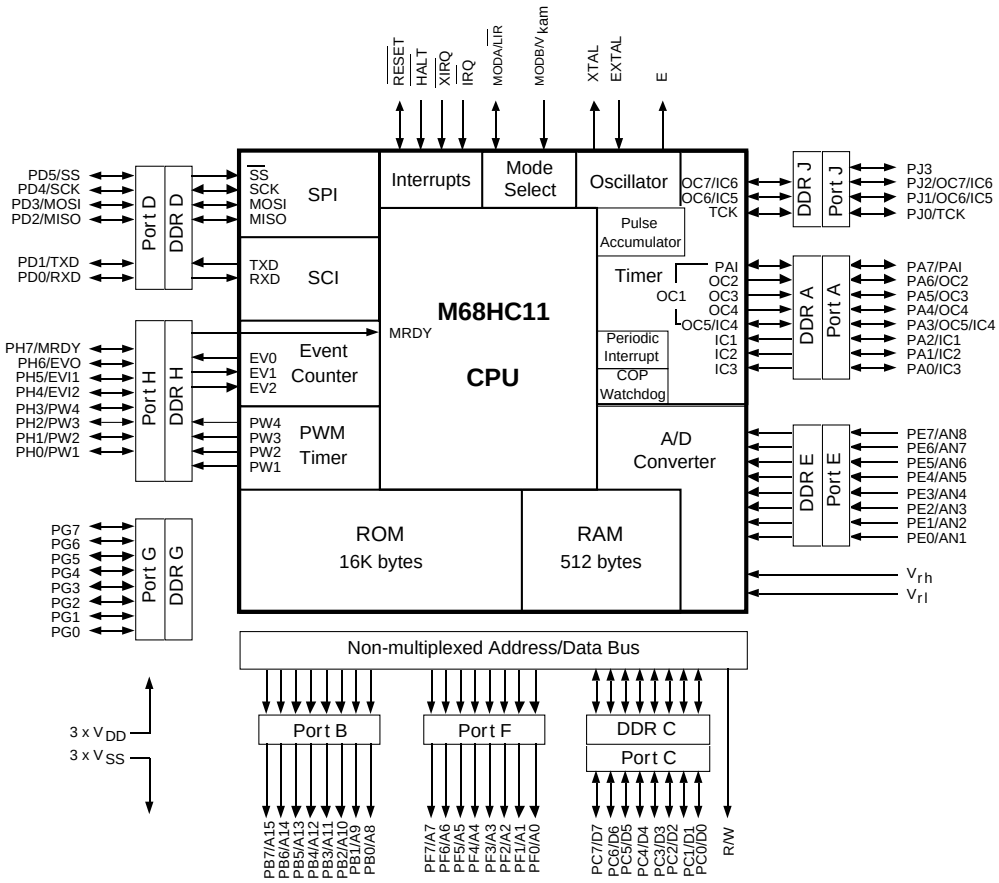


Figure 1-1. Functional Block Diagram

SECTION 2

OPERATING MODES AND SIGNAL DESCRIPTION

This section describes the operating modes and signals of the MC68HC11G5 MCU.

2.1 OPERATING MODES

During reset the MC68HC11G5 uses two mode select pins, MODA and MODB, to select one of two normal modes or one of two special operating modes. The normal operating modes are the single chip mode, which allows maximum use of the pins for on-chip peripheral functions, and the expanded non-multiplexed mode which allows access to the 64 kbytes of memory space including the internal RAM, ROM (if present), and register spaces. The special operating modes are the bootstrap mode (which causes all vectors to be fetched from the on-chip bootstrap ROM), and the test mode (which is an expanded mode allowing special testing functions on the MCU subsystems). The special operating modes have access to certain privileged control bits which are not available in the normal operating modes.

Mode selection, according to the values encoded on the mode select pins (MODA and MODB), is shown in Table 2-1. The states of MODA and MODB, captured during reset determine the logic state of the Special Mode (SMOD) and the Mode A Select (MDA) control bits in the HPRIO register.

Table 2-1. Mode Select Summary

Input Pins			Latched at Reset	
MODA	MODB	Mode Selected	SMOD	MDA
0	1	Single Chip	0	0
1	1	Expanded Non-Multiplexe(	0	1
0	0	Bootstrap	1	0
1	0	Test	1	1

2.1.1 Single Chip Operating Mode

In single chip mode, the MCU functions as a self-contained microcontroller and has no external address or data bus. This mode provides maximum use of the pins for on-chip peripheral functions (66 pins). In single chip mode the R/$\overline{W}$ line is always high (read) to facilitate changing mode.

2.1.2 Expanded Non-Multiplexed Operating Mode

In the expanded non-multiplexed mode, the MCU can address up to 64 kbytes of address space. High order address bits are output on Port B and low order address bits are output on Port F. The bidirectional data bus is on Port C and the R/$\overline{W}$ pin is used to control the direction of data transfer on this bus.

2.1.3 Bootstrap Operating Mode

This is a very versatile operating mode since there are essentially no limitations on the special purpose program that can be loaded into internal RAM. The resident bootstrap program, contained in the bootstrap ROM, uses the SCI to read a variable length program into on-chip RAM. Program control is passed to RAM at location $0000 when an idle line of at least four characters occurs.

The MC68HC11G5 communicates through the SCI port. After reset in bootstrap mode, the SCI runs at E/16 (7812 baud for E-clock = 2 MHz). A break condition is output on the SCI transmitter. For normal use of the bootstrap program, the user must send $FF to the SCI receiver at either E/16 or E/104 (1200 baud for E-clock = 2 MHz).

Note: The $FF is not echoed through the SCI transmitter.

In this mode all interrupt vectors are mapped to pseudo-vectors in RAM (refer to Table 2-2). This allows programmers to use their own service-routine addresses. Each pseudo-vector is allowed three bytes of space, rather than the two bytes for normal vectors, because an explicit jump (JMP) opcode is needed to cause the desired jump to the user's service-routine address.

Since the SMOD control bit is initialized to one, all of the privileged control bits are accessible in this mode. This allows the bootstrap mode to be used for test and diagnostic functions on completed modules. Mode switching can occur under program control by writing to the SMOD and MDA bits of the HPRIO register.

2.1.4 Test Operating Mode

This special expanded mode is primarily intended for factory testing. In this mode the reset and interrupt vectors are fetched externally from locations $BFFE – $BFFF. The SMOD bit in the HPRIO register will be set, indicating that a special mode is in effect and allowing the user to write to certain privileged control bits which are normally read-only bits. Also, a special register, TEST1, is enabled which allows several factory test functions to be invoked.

It is also possible to change from the test mode to a normal operating mode by writing a zero to the SMOD bit. Once this bit is cleared, it cannot be changed back to one without going through a reset sequence.

Table 2-2. Bootstrap Mode Jump Vectors

Address	Pseudo-vector
00B5	Event 2
00B8	Event 1
00BB	Timer Overflow 2
00BE	Timer Output Compare 7 / Input Capture
00C1	Timer Output Compare 6 / Input Capture
00C4	SCI
00C7	SPI
00CA	Pulse Accumulator Input Edg
00CD	Pulse Accumulator Overflow
00D0	Timer Overflow 1
00D3	Timer Output Compare 5 / Input Capture
00D6	Timer Output Compare 4
00D9	Timer Output Compare 3
00DC	Timer Output Compare 2
00DF	Timer Output Compare 1
00E2	Timer Input Capture 3
00E5	Timer Input Capture 2
00E8	Timer Input Capture 1
00EB	Real Time Interrup
00EE	IRQ
00F1	XIRQ
00F4	SWI
00F7	Illegal Op-cod
00FA	COP Fai
00FD	Clock Monitor
BF40	Reset

2.1.4.1 Factory Test Register (TEST1)

	7	6	5	4	3	2	1	0	
$103E	TILOP	TPWSL	OCCR	CBYP1	DISR	FCM	FCOP	CBYP2	TEST1
RESET:	0	0	0	0	0	0	0	0	

READ: Any time (always zero if not in a test mode)

WRITE: Only while SMOD = 1 (TEST or BOOT modes)

TILOP — Test Illegal Opcode

0 – Normal operation (trap on illegal opcodes)

1 – Stop giving LIR when an illegal opcode is found

This test mode allows testing of all illegal opcodes serially without servicing an interrupt after each illegal opcode is received. $\overline{\text{LIR}}$ (inactive for illegal opcodes) is used to monitor correct CPU operation.

TPWSL — Pulse Width Modulation Scaled Clock

0 – Normal operation

1 – Clock S (Scaled) from the PWM timer is output to and readable on the PWSCAL register. Normal writing of the PWSCAL register will still function.

OCCR — Output Condition Code Register Status to Timer Port

0 – Normal operation

1 – The Condition Code bits H, N, Z, V, and C are driven out of the 5 most significant bits (bits 3 – 7, respectively) of the timer I/O port to allow the test system to monitor CPU operation.

CBYP1 — Timer Counter 1 Divider Chain By-pass

0 – Normal operation

1 – 16-bit free running timer counter 1 is divided into 8-bit halves and the prescaler is by-passed. The system E-clock drives both halves directly.

DISR — Disable Resets from COP and Clock Monitor

0 – Normal operation

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

When the device is reset in test or bootstrap mode (SMOD = 1), the DISR bit is initialized to 1 so that resets from the COP and clock monitor are disabled. When reset in normal modes, the DISR bit is initialized to 0.

FCM — Force Clock Monitor Failure

0 – Normal operation

1 – Immediately generates a Clock Monitor failure reset. Note that the CME bit must also be set for the reset to be forced.

FCOP — Force COP Watchdog Failure

0 – Normal operation

1 – Immediately generates a COP failure reset. Note that COP must be otherwise enabled for the reset to be generated; NOCOP = 0.

CBYP2 — Timer Counter 2 Chain By-pass

0 – Normal operation

1 – 16-bit free running timer counter 2 is divided into 8-bit halves and the prescaler is by-passed. The system E-clock drives both halves directly.

The DISR control bit has priority over the FCM and FCOP control bits such that, if DISR is set to one, no reset results even when FCM or FCOP is set to one.

2.2 SIGNAL DESCRIPTION

The following table shows the pin usage for the MC68HC11G5.

5 Volt Supply (V_{DD}, V_{DDL}, V_{DDR})	3
Ground (V_{SS}, V_{SSL}, V_{SSR})	3
$\overline{\text{RESET}}$	1
Oscillator (XTAL, EXTAL)	2
E	1
R/$\overline{\text{W}}$	1
$\overline{\text{IRQ}}$	1
$\overline{\text{XIRQ}}$	1
$\overline{\text{HALT}}$	1
Mode Select (V_{kam} & $\overline{\text{LIR}}$)	2
Analog Reference (V_{rh}, V_{rl})	2
8-bit Ports (7)	56
6-bit Port (1)	6
4-bit Port (1)	4
Total	84 pins

The following paragraphs describe the input/output signals used by the various functions of the MCU.

2.2.1 Input Power (V_{DD}) and Ground (V_{SS})

Power is supplied to the MCU via three positive supply pins (V_{DD}) and three ground pins (V_{SS}). Note that, for ease of identification, the V_{DD} pins on the left and right sides of the device package are called V_{DDL} and V_{DDR}, respectively. Similarly, the V_{SS} pins on the left and right sides of the device package are called V_{SSL} and V_{SSR}.

2.2.2 Reset ($\overline{\text{RESET}}$)

This active low bidirectional control signal pin is used as an input to initialize the MC68HC11G5 to a known start-up state, and as an open-drain output to indicate an internal computer operating properly (COP) watchdog circuit or clock monitor failure. This reset signal is significantly different from the reset signal used on other Motorola MCUs. Refer to **SECTION 5: RESETS, INTERRUPTS AND LOW POWER MODES** before designing circuitry to generate or monitor this signal.

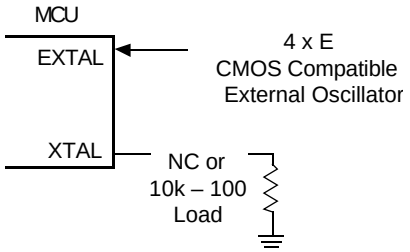
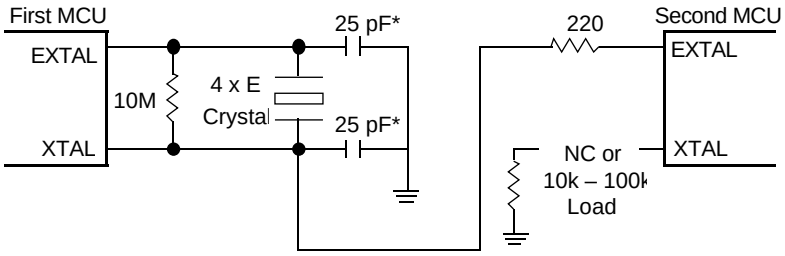
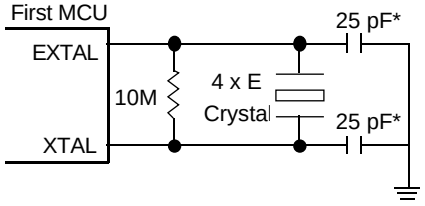


Figure 2-1. Oscillator circuits: (a) External oscillator connections



* This value contains all stray capacitance

Figure 2-1. Oscillator circuits: (b) One crystal driving two MCUs



* This value contains all stray capacitance

Figure 2-1. Oscillator circuits: (c) Common crystal connections

2.2.3 Crystal Driver (XTAL) and External Clock Input (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. When a crystal is used, a 25 pF capacitor should be connected from each of the XTAL and EXTAL pins to ground to ensure reliable start-up operation. When an external CMOS compatible clock is used as an input to the EXTAL pin, the XTAL pin should be left disconnected. However, a 10 kΩ to 100 kΩ load resistor to ground may be used to reduce RFI noise emission. (See Figure 2-1.)

2.2.4 E-clock Output (E)

The E-clock pin provides an output for the internally generated E-clock, which can be used as a timing reference. The frequency of the E-clock output is one quarter of the input frequency at the XTAL and EXTAL pins. When the E-clock output is low, an internal process is taking place; when high, data is being accessed. The signal is halted when the MCU is in STOP mode.

2.2.5 Read/Write (R/$\overline{W}$)

This pin is used to control the direction of transfers on the external data bus in expanded non-multiplexed mode. A logic zero level indicates that data is being written to the external data bus; a logic one level indicates that a read cycle is in progress. R/$\overline{W}$ stays high during single-chip and bootstrap modes to maintain the ‘read’ state for systems which switch modes.

2.2.6 Interrupt Request ($\overline{IRQ}$)

The $\overline{IRQ}$ pin provides the capability to apply asynchronous interrupts to the MC68HC11G5. It is software selectable, (using the IRQE bit in the OPTION register) with a choice of either negative edge sensitive or level sensitive triggering, and is always configured to level sensitive by reset. This pin requires an external pull-up resistor connected to V_{DD}.

2.2.7 Non-Maskable Interrupt ($\overline{XIRQ}$)

The $\overline{XIRQ}$ pin provides the capability for applying asynchronous non-maskable interrupts, after reset initialization, to the MC68HC11G5. During reset, the X-bit in the condition code register is set and any interrupt is masked until the MCU software enables it. The $\overline{XIRQ}$ input is level sensitive, and requires an external pull-up resistor to V_{DD}.

2.2.8 Halt ($\overline{HALT}$)

This pin is used as a clean way to force the processor bus into a tri-state condition by means of an external signal. When $\overline{HALT}$ is pulled low the processor completes execution of the present instruction and then tri-states the address and data bus. When the $\overline{HALT}$ pin is released, the processor continues normal execution.

The $\overline{HALT}$ pin can also be used as a “clean way” to put the part into reset. Once pulled into halt, $\overline{RESET}$ can be pulled low, ensuring that the processor will be reset on an instruction boundary rather than in the middle of an instruction.

2.2.9 Mode A/Load Instruction Register (MODA/$\overline{\text{LIR}}$) and Mode B/Standby Voltage (MODB/V_{kam})

During reset, MODA and MODB are used to select one of the four basic operating modes. Refer to Table 2-1 for mode selection and to paragraph **2.1: OPERATING MODES** for additional details.

After the operating mode has been selected, the open drain $\overline{\text{LIR}}$ pin goes to an active low level during the first E-clock cycle of each instruction. The $\overline{\text{LIR}}$ pin can be used as an aid to program debugging.

The V_{kam} signal is used as the input for RAM standby power and will retain the RAM contents during power down.

2.2.10 A/D Converter Reference Voltages (V_{rl}, V_{rh})

These two pins provide the reference high and low voltages to the internal Analog-to-Digital converter circuitry. Refer to **SECTION 9: ANALOG-TO-DIGITAL CONVERTER**.

2.2.11 Port Signals

The 66 input/output (I/O) lines are arranged into seven 8-bit ports (ports A, B, C, E, F, G and H), one 6-bit port (Port D) and one 4-bit port (Port J). Most of these ports serve more than one purpose, depending on the operating mode or peripheral functions selected. Table 2-3 shows the different functions of each port pin in the four operating modes. For more detailed information on I/O ports refer to **Section 4: INPUT/OUTPUT PORTS.**

Table 2-3. Port Signal Functions: (a) Expanded Non-multiplexed and Test Modes

	Port								
Bit	**A**	**B**	**C**	**D**	**E**	**F**	**G**	**H**	**J**
0	PA0/IC3	A8	D0	PD0/RXD	PE0/AN1	A0	PG0	PH0/PW1	PJ0/TCK
1	PA1/IC2	A9	D1	PD1/TXD	PE1/AN2	A1	PG1	PH1/PW2	PJ1/IC5/OC6
2	PA2/IC1	A10	D2	PD2/MISO	PE2/AN3	A2	PG2	PH2/PW3	PJ2/IC6/OC7
3	PA3/IC4/OC5 (and/or OC1)	A11	D3	PD3/MOSI	PE3/AN4	A3	PG3	PH3/PW4	PJ3
4	PA4/OC4 (and/or OC1)	A12	D4	PD4/SCK	PE4/AN5	A4	PG4	PH4/EVI2	
5	PA5/OC3 (and/or OC1)	A13	D5	PD5/$\overline{SS}$	PE5/AN6	A5	PG5	PH5/EVI1	
6	PA6/OC2 (and/or OC1)	A14	D6		PE6/AN7	A6	PG6	PH6/EVO	
7	PA7/PA1 (and/or OC1)	A15	D7		PE7/AN8	A7	PG7	PH7/MRDY	

Table 2-3. Port Signal Functions: (b) Single Chip and Bootstrap Modes

	Port								
Bit	**A**	**B**	**C**	**D**	**E**	**F**	**G**	**H**	**J**
0	PA0/IC3	PB0	PC0	PD0/RXD	PE0/AN1	PF0	PG0	PH0/PW1	PJ0/TCK
1	PA1/IC2	PB1	PC1	PD1/TXD	PE1/AN2	PF1	PG1	PH1/PW2	PJ1/IC5/OC6
2	PA2/IC1	PB2	PC2	PD2/MISO	PE2/AN3	PF2	PG2	PH2/PW3	PJ2/IC6/OC7
3	PA3/IC4/OC5 (and/or OC1)	PB3	PC3	PD3/MOSI	PE3/AN4	PF3	PG3	PH3/PW4	PJ3
4	PA4/OC4 (and/or OC1)	PB4	PC4	PD4/SCK	PE4/AN5	PF4	PG4	PH4/EVI2	
5	PA5/OC3 (and/or OC1)	PB5	PC5	PD5/$\overline{SS}$	PE5/AN6	PF5	PG5	PH5/EVI1	
6	PA6/OC2 (and/or OC1)	PB6	PC6		PE6/AN7	PF6	PG6	PH6/EVO	
7	PA7/PA1 (and/or OC1)	PB7	PC7		PE7/AN8	PF7	PG7	PH7	

This page intentionally left blank

SECTION 3

MEMORY AND CONTROL/STATUS REGISTERS

This section describes the memory, memory subsystems mapping, and the mapping of the control and status registers of the MC68HC11G5 MCU.

3.1 ROM

The internal 16 kilobytes of ROM occupy the highest 16k addresses in the memory map ($C000 – $FFFF). On coming out of reset, this ROM is enabled in the single chip, test and bootstrap modes. This ROM is disabled when the ROMON bit in the CONFIG register is clear. This register bit is only writable in the special modes, bootstrap and test. For normal single chip mode, the ROM is on (ROMON = 1). In expanded mode the ROM is off (ROMON = 0) thus allowing the user to execute a program in the external memory space. (See also **SECTION 3.4.5: CHANGING MODES.**)

3.2 BOOTSTRAP ROM

The 256 bytes of bootstrap ROM are located at memory locations $BF00 – $BFFF. The reset and interrupt vectors, while in this mode, are addressed at $BFC0 – $BFFF. The interrupt vectors point to pseudo-vectors located in RAM (see Section 2.1.3 and Table 2-2).

3.3 RAM

Using the INIT control register, the 512 byte block of static RAM is mappable to the start of any 4 kilobyte boundary in memory. The reset default position of the RAM is located at $0000 to $01FF. If the internal registers are mapped in the same 4 kilobyte block as RAM, the registers will have higher priority.

The RAM is implemented with static cells and retains its contents during WAIT, HALT and STOP modes. The contents of the RAM can also be retained during power-down, by supplying a low current back-up power source to the V_{kam} pin (MODB).

3.4 MEMORY MAP

Each of the normal and special operating modes of the MC68HC11G5 has a default initial memory map. In addition, there is a control register (CONFIG) which can be used to remove (disable) the ROM from the memory map. After reset the INIT register (which is reset to $01 independent of mode and may only be written under specific circumstances) can alter the mapping of RAM and internal I/O resources under software control.

While in bootstrap mode, the restart and interrupt vectors are fetched from an alternate memory area ($BFC0 – $BFFF) in an internal bootstrap ROM. In the test mode, vectors are also fetched from the alternate area ($BFC0 – $BFFF), however, since the bootstrap ROM is not enabled, these vectors reside in external memory.

Figure 3-1 shows the memory maps for all four modes of operation: single chip, expanded non-multiplexed, bootstrap and test. On-board memory locations are shown by the shaded areas and the contents of these areas are described on the right.

3.4.1 Single Chip Mode

This normal operating mode is established by having a logic one level on the MODB/V_{kam} pin and a logic zero level on the MODA/$\overline{LIR}$ pin at the rising edge of $\overline{RESET}$. The ROMON bit in the CONFIG register is a one out of reset in this mode, so that the user ROM appears in the map. The initial memory map which results is also controlled by the state of three bits (RBOOT, SMOD, and MDA) in the HPRIO control register which are initialized automatically by hardware prior to the rising edge on $\overline{RESET}$.

Initial conditions affecting the memory map are:

ROMON = 1, RBOOT = 0, SMOD = 0, MDA = 0

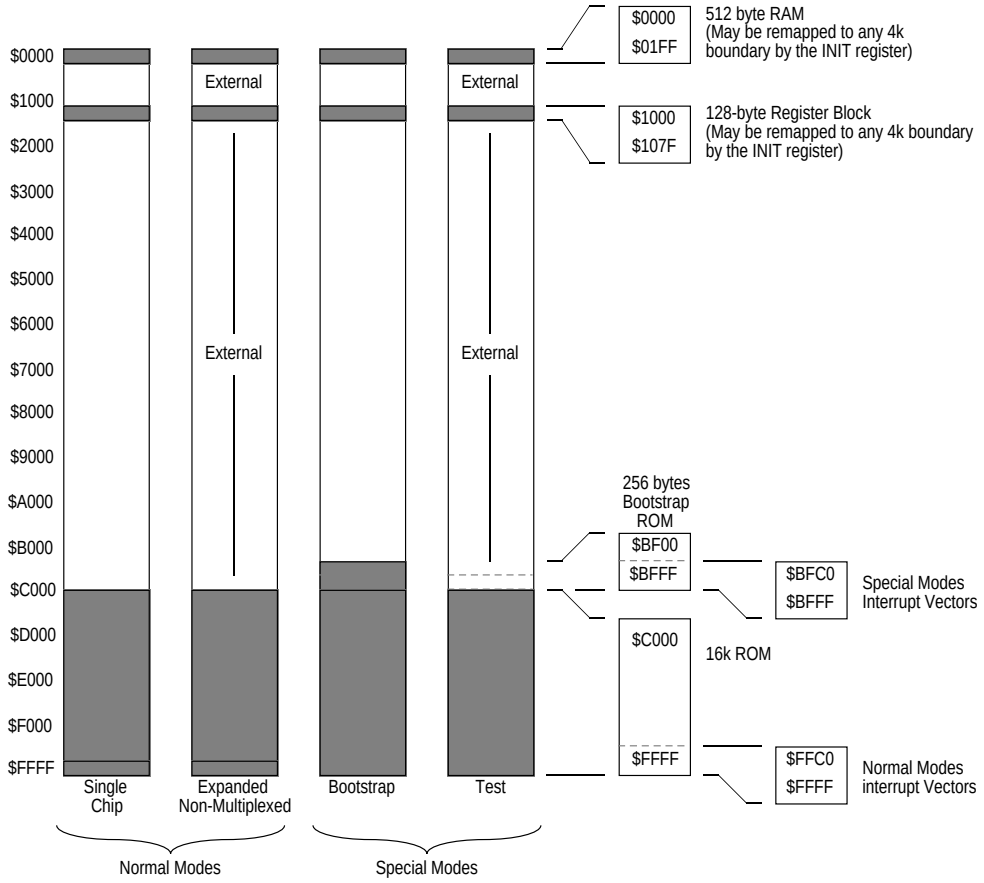


Figure 3-1. Memory Map

3.4.2 Expanded Non-multiplexed Mode

This normal operating mode is established by having a logic one level on both the MODB/V_{kam} pin and the MODA/$\overline{LIR}$ pin at the rising edge of $\overline{RESET}$. The ROMON bit in the CONFIG register is a zero in this mode so that the user ROM does **not** appear in the map. The initial memory map which results is also controlled by the state of three bits (RBOOT, SMOD, and MDA) in the HPRIO control register which are initialized automatically by hardware prior to the rising edge on $\overline{RESET}$.

Initial conditions affecting the memory map are:

ROMON = 0, RBOOT = 0, SMOD = 0, MDA = 1

3.4.3 Special Bootstrap Mode

This special mode is established by having a logic zero level on both the MODB/V_{kam} pin and the MODA/$\overline{LIR}$ pin at the rising edge of $\overline{RESET}$. The ROMON bit in the CONFIG register is a one out of reset in this mode, so that the user ROM appears in the map. The initial memory map which results is also controlled by the state of three bits (RBOOT, SMOD, and MDA) in the HPRIO control register which are initialized automatically by hardware prior to the rising edge on $\overline{RESET}$.

Initial conditions affecting the memory map are:

ROMON = 1, RBOOT = 1, SMOD = 1, MDA = 0

3.4.4 Special Test Mode

This special mode is established by having a logic zero level on the MODB/V_{kam} pin and a logic one level on the MODA/$\overline{LIR}$ pin at the rising edge of $\overline{RESET}$. The ROMON bit in the CONFIG register is a one out of reset in this mode, so that the user ROM appears in the map. The initial memory map which results is actually controlled by the state of three bits (RBOOT, SMOD, and MDA) in the HPRIO control register which were initialized automatically by hardware prior to the rising edge on $\overline{RESET}$.

Initial conditions affecting the memory map are:

ROMON = 1, RBOOT = 0, SMOD = 1, MDA = 1

3.4.5 Changing Modes

Changing the operating mode and the bits described above is subject to some restrictions in normal modes. First of all, ROMON, RBOOT and SMOD will stay in their reset state. These bits cannot be changed in the normal operating modes. In order to allow a user to have an expanded mode system with the ROM on, the user is allowed to change MDA once only. This means that the user can bring the device out of reset in single chip mode, and then write MDA to a 1, thus putting the part into expanded mode with the ROM on (ROMON = 1).

If the user does not intend to change modes after coming out of reset, he may wish to write MDA one time to the same state. As MDA may be written to only once; this will prevent an accidental write from changing the mode at some later time.

In the special operating modes, ROMON, RBOOT and MDA may be written at any time to either 0 or 1. SMOD cannot be written back to a 1 after being written to a 0, because the part will no longer be in a special mode.

3.5 SYSTEM CONFIGURATION

The MC68HC11G5 allows the user to configure the MCU system to his specific requirements via hard wired options, such as the mode select pins, and via internal software programmable control

registers. Two special internal registers (INIT and CONFIG) require further explanation. The INIT control register allows the RAM and internal register block to be repositioned in the memory map during software initialization. The CONFIG control register controls the presence of ROM in the memory map as well as the NOCOP watch-dog system enable.

3.5.1 RAM and I/O Mapping Register (INIT)

The INIT register is a special purpose 8-bit register that is used during initialization to change the default locations of 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, and thereafter becomes a read-only register.

	7	6	5	4	3	2	1	0	
$103D	RAM3	RAM2	RAM1	RAM0	REG3	REG2	REG1	REG0	INIT
RESET:	0	0	0	0	0	0	0	1	

READ: Any time

WRITE: If SMOD = 0, may be written once only, and only during the first 64 cycles: after this time the register becomes read-only. If SMOD = 1 (test or bootstrap mode), writes are always permitted.

RAM3, RAM2, RAM1, RAM0 — Internal RAM map position

These four bits specify the four most significant bits of the 16-bit RAM address.

REG3, REG2, REG1, REG0 — Register block map position.

These four bits specify the four most significant bits of the 16-bit internal register space address. For more details refer to the memory map.

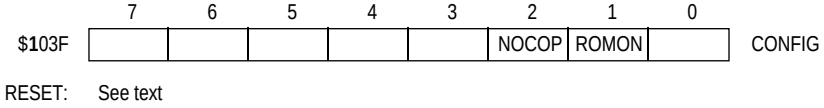
Since the INIT register is set to $01 by reset, the default starting address is $0000 for RAM and $1000 for the 128 byte control and status register block. The upper four bits of the INIT register specify the starting address for the 512 byte RAM, while the lower four bits specify the starting address for the control and status register block. In each case, the four bits define the four most significant bits of the 16-bit address.

If relocation conflicts occur between the RAM and the ROM, the RAM takes priority over the ROM which simply becomes inaccessible at those locations. Similarly, if the control and status registers are located so that they conflict with the RAM and/or ROM, the registers take priority and the RAM and/or ROM at those locations become inaccessible. Also, if an internal resource conflicts with an external device, no harmful conflict occurs. (Data from the external device is not applied to the internal data bus, thus it cannot interfere with the internal read.)

Note: There are unused register locations in the 128 byte control and status register block. Reads of these unused registers return data from the undriven internal data bus, not from another source that happens to be located at the same address.

3.5.2 Configuration Control Register (CONFIG)

The CONFIG control register is used for system configuration control functions including removal of ROM from the memory map, and enabling of the COP watch-dog system.



NOCOP — COP System Disable

READ: Any time

WRITE: If SMOD = 0, may be written once only, and only during the first 64 cycles: after this time the bit becomes read-only. If SMOD = 1 (test or bootstrap mode), writes are always permitted.

RESET: 0 (COP on) for SMOD = 0; 1 (COP off) for SMOD = 1

0 – COP system enabled (forces reset on timeout)

1 – COP system does not force reset on timeout

ROMON — ROM Enable

READ: Any time

WRITE: May not be written if SMOD = 0. If SMOD = 1, bit is writable at any time.

RESET: 1 (ROM on) for single chip, bootstrap and test modes; 0 (ROM off) for normal expanded mode

1 – ROM is present in the memory map (at $C000 – $FFFF)

0 – ROM is disabled from the memory map

3.6 CONTROL/STATUS REGISTERS

The control and status registers used to control the operation of the MCU are contained in a 128 byte block which can be relocated to any 4 kilobyte boundary in memory. Throughout this document, control and status register addresses are displayed with the high order digit shown as a bold "**1**" to indicate that the register block may be relocated to some 4 kilobyte memory page other than its default position of $**1**000. Figure 3-2 is provides a complete list of the control and status registers and reserved locations in this block of memory.

$1000	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0	PORTA	I/O Port A
$1001	DDA7	DDA6	DDA5	DDA4	DDA3	DDA2	DDA1	DDA0	DDRA	Data Direction for Port A
$1002	PG7	PG6	PG5	PG4	PG3	PG2	PG1	PG0	PORTG	I/O Port G
$1003	DDG7	DDG6	DDG5	DDG4	DDG3	DDG2	DDG1	DDG0	DDRG	Data Direction for Port G
$1004	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0	PORTB	I/O Port B
$1005	PF7	PF6	PF5	PF4	PF3	PF2	PF1	PF0	PORTF	I/O Port F
$1006	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0	PORTC	I/O Port C
$1007	DDC7	DDC6	DDC5	DDC4	DDC3	DDC2	DDC1	DDC0	DDRC	Data Direction for Port C
$1008	0	0	PD5	PD4	PD3	PD2	PD1	PD0	PORTD	I/O Port D
$1009	0	0	DDD5	DDD4	DDD3	DDD2	DDD1	DDD0	DDRD	Data Direction for Port D
$100A	PE7	PE6	PE5	PE4	PE3	PE2	PE1	PE0	PORTE	I/O Port E
$100B	FOC1	FOC2	FOC3	FOC4	FOC5	FOC6	FOC7	0	CFORC	Compare Force Register
$100C	OC1M7	OC1M6	OC1M5	OC1M4	OC1M3	0	0	0	OCIM	OC1 Action Mask Register
$100D	OC1D7	OC1D6	OC1D5	OC1D4	OC1D3	0	0	0	OCID	OC1 Action Data Register
$100E	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	TCNT1	Timer Counter Register 1
$100F	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0		
$1010	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	TIC1	Input Capture 1 Register
$1011	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0		
$1012	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	TIC2	Input Capture 2 Register
$1013	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0		
$1014	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	TIC3	Input Capture 3 Register
$1015	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0		
$1016	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	TOC1	Output Compare 1 Register
$1017	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0		
$1018	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	TOC2	Output Compare 2 Register
$1019	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0		
$101A	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	TOC3	Output Compare 3 Register
$101B	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0		
$101C	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	TOC4	Output Compare 4 Register
$101D	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0		
$101E	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	TO5I4	Output Compare 5/ Input Capture 4 Register
$101F	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0		

Figure 3-2. Control and Status Registers (Page 1)

$1020	OM2	OL2	OM3	OL3	OM4	OL4	OM5	OL5	TCTL1	Timer Control Register 1
$1021	EDG4B	EDG4A	EDG1B	EDG1A	EDG2B	EDG2A	EDG3B	EDG3A	TCTL2	Timer Control Register 2
$1022	OC1I	OC2I	OC3I	OC4I	4/5I	IC1I	IC2I	IC3I	TMSK1	Main Timer Interrupt Mask Register 1
$1023	OC1F	OC2F	OC3F	OC4F	4/5F	IC1F	IC2F	IC3F	TFLG1	Main Timer Interrupt Flag Register 1
$1024	TO1I	RTII	PAOVI	PAII	TO2I	5/6I	6/7I	0	TMSK2	Misc. Timer Interrupt Mask Register 2
$1025	TO1F	RTIF	PAOVF	PAIF	TO2F	5/6F	6/7F	0	TFLG2	Misc. Timer Interrupt Flag Register 2
$1026	0	PAEN	PAMOD	PEDGE	I4/O5	RTR2	RTR1	RTR0	PACTL	Pulse Accumulator Control Register
$1027	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	PACNT	Pulse Accumulator Count Register
$1028	SPIE	SPE	DWOM	MSTR	CPOL	CPHA	SPR1	SPR0	SPCR	SPI Control Register
$1029	SPIF	WCOL	0	MODF	0	0	0	0	SPSR	SPI Status Register
$102A	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	SPDAT	SPI Data Register
$102B	TCLR	0	SCP1	SCP0	RCKB	SCR2	SCR1	SCR0	BAUD	SCI Baud Rate Control
$102C	R8	T8	0	M	WAKE	0	0	0	SCCR1	SCI Control Register 1
$102D	TIE	TCIE	RIE	ILIE	TE	RE	RWU	SBK	SCCR2	SCI Control Register 2
$102E	TDRE	TC	RDRF	IDLE	OR	NF	FE	0	SCSR	SCI Status Register
$102F	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	SCDAT	SCI Data (Read RDR, Write TDR)
$1030	CCF	CONV8	SCAN	MULT	CD	CC	CB	CA	ADCTL	A/D Control Register
$1031	0	0	0	0	PJ3	PJ2	PJ1	PJ0	PORTJ	I/O Port J
$1032	0	0	0	0	DDJ3	DDJ2	DDJ1	DDJ0	DDRJ	Data Direction for Port J
$1033	PH7	PH6	PH5	PH4	PH3	PH2	PH1	PH0	PORTH	I/O Port H
$1034	DDH7	DDH6	DDH5	DDH4	DDH3	DDH2	DDH1	DDH0	DDRH	Data Direction for Port H
$1035	0	0	0	0	0	0	0	0	Reserved	
$1036	0	0	0	0	0	0	0	0	Reserved	
$1037	0	0	0	0	0	0	0	0	Reserved	
$1038	GWOM	CWOM	0	IRV	0	0	MRDY	NHALT	OPT2	System Configuration Options 2 Reg.
$1039	ADPU	CSEL	IRQE	DLY	CME	0	CR1	CR0	OPTION	System Configuration Options
$103A	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	COPRST	Arm/Reset COP Timer Circuitry
$103B	0	0	0	0	0	0	0	0	Reserved	
$103C	RBOOT	SMOD	MDA	PSEL4	PSEL3	PSEL2	PSEL1	PSEL0	HPRIO	Highest Priority I-Bit Interrupt and Misc.
$103D	RAM3	RAM2	RAM1	RAM0	REG3	REG2	REG1	REG0	INIT	RAM and I/O Mapping Register
$103E	TILOP	TPWSL	OCCR	CBYP1	DISR	FCM	FCOP	CBYP2	TEST1	Factory Test Control Register
$103F	0	0	0	0	0	NOCOP	ROMON	0	CONFIG	COP and ROM Enables

Figure 3-2. Control and Status Registers (Page 2)

$1040	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	ADR1	A/D Result Register 1
$1041	Bit 7	Bit 6	0	0	0	0	0	0		
$1042	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	ADR2	A/D Result Register 2
$1043	Bit 7	Bit 6	0	0	0	0	0	0		
$1044	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	ADR3	A/D Result Register 3
$1045	Bit 7	Bit 6	0	0	0	0	0	0		
$1046	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	ADR4	A/D Result Register 4
$1047	Bit 7	Bit 6	0	0	0	0	0	0		
$1048	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	ADR5	A/D Result Register 5
$1049	Bit 7	Bit 6	0	0	0	0	0	0		
$104A	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	ADR6	A/D Result Register 6
$104B	Bit 7	Bit 6	0	0	0	0	0	0		
$104C	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	ADR7	A/D Result Register 7
$104D	Bit 7	Bit 6	0	0	0	0	0	0		
$104E	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	ADR8	A/D Result Register 8
$104F	Bit 7	Bit 6	0	0	0	0	0	0		
$1050	EDG5B	EDG5A	EDG6B	EDG6A	OM6	OL6	OM7	OL7	TCTL3	Timer Control Register 3
$1051	CT2SP	CT1SP	0	0	0	0	I5/O6	I6/O7	TCTL4	Timer Control Register 4
$1052	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	TCNT2	Timer Counter Register 2
$1053	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0		
$1054	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	TO6I5	Output Compare 6/ Input Capture 5 Register
$1055	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0		
$1056	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	TO7I6	Output Compare 7/ Input Capture 6 Register
$1057	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0		
$1058	TEDGB	TEDGA	PR2B	PR2A	0	0	PR1B	PR1A	TPRE	Timer Prescaler Register
$1059	0	0	0	0	0	0	0	0	Reserved	
$105A	0	0	0	0	0	0	0	0	Reserved	
$105B	0	0	0	0	0	0	0	0	Reserved	
$105C	0	0	0	0	0	0	0	0	Reserved	
$105D	0	0	0	0	0	0	0	0	Reserved	
$105E	0	0	0	0	0	0	0	0	Reserved	
$105F	0	0	0	0	0	0	0	0	Reserved	

Figure 3-2. Control and Status Registers (Page 3)

$1060	CON34	CON12	PCKA2	PCKA1	0	PCKB3	PCKB2	PCKB1	PWCLK	PWM Timer Clock Select
$1061	PCLK4	PCLK3	PCLK2	PCLK1	PPOL4	PPOL3	PPOL2	PPOL1	PWPOL	PWM Timer Polarity
$1062	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	PWSCAL	PWM Timer Prescaler
$1063	0	0	0	0	PWEN4	PWEN3	PWEN2	PWEN1	PWEN	PWM Timer Enable
$1064	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	PWCNT1	PWM Timer Counter 1
$1065	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	PWCNT2	PWM Timer Counter 2
$1066	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	PWCNT3	PWM Timer Counter 3
$1067	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	PWCNT4	PWM Timer Counter 4
$1068	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	PWPER1	PWM Timer Period 1
$1069	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	PWPER2	PWM Timer Period 2
$106A	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	PWPER3	PWM Timer Period 3
$106B	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	PWPER4	PWM Timer Period 4
$106C	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	PWDTY1	PWM Timer Duty 1
$106D	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	PWDTY2	PWM Timer Duty 2
$106E	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	PWDTY3	PWM Timer Duty 3
$106F	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	PWDTY4	PWM Timer Duty 4
$1070	0	0	EVMDB	EVMDA	0	EVCKC	EVCKB	EVCKA	EVCLK	Event Counter Clock Select
$1071	EVOEN	EVPOL	EVI2C	EVI2B	EVI2A	EVI1C	EVI1B	EVI1A	EVCTL	Event Counter Control Register
$1072	EVCEN	0	0	0	0	0	EV2I	EV1I	EVMSK	Event Counter Interrupt Mask Register
$1073	0	0	0	0	0	0	EV2F	EV1F	EVFLG	Event Counter Interrupt Flag Register
$1074	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	EVCNT1	Event Counter Count Register 1
$1075	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	EVCNT2	Event Counter Count Register 2
$1076	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	ECMP1A	Event Counter Compare Register 1A
$1077	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	ECMP2A	Event Counter Compare Register 2A
$1078	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	ECMP1B	Event Counter Compare Register 1B
$1079	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	ECMP2B	Event Counter Compare Register 2B
$107A	0	0	0	0	0	0	0	0	Reserved	
$107B	0	0	0	0	0	0	0	0	Reserved	
$107C	0	0	0	0	0	0	0	0	Reserved	
$107D	0	0	0	0	0	0	0	0	Reserved	
$107E	0	0	0	0	0	0	0	0	Reserved	
$107F	0	0	0	0	0	0	0	0	Reserved	

Figure 3-2. Control and Status Registers (Page 4)

SECTION 4

INPUT/OUTPUT PORTS

There are seven 8-bit I/O ports, one 6-bit I/O port and one 4-bit I/O port on the MC68HC11G5. Most of the 66 I/O pins serve multiple purposes depending on the configuration of the MCU system (see Table 2-3). The configuration is controlled in turn, by hardware mode selection as well as by several internal control registers.

Ports A, C, D, G, H, and J may be used as general purpose input and/or output pins as specified by DDRA, DDRC, DDRD, DDRG, DDRH, and DDRJ. Ports B, E, and F have fixed data direction and thus do not require DDR control registers.

Port signals can always be sensed (read) even if they are configured as general purpose outputs or are dedicated to on-chip peripheral functions. Reading the port address returns the sensed logic level at the pin when the pin is configured as an input, and the logic sense at the input to the pin driver is sensed when the pin is configured as an output.

I/O pins default to being general purpose I/O lines unless an internal function which uses that pin is specifically enabled. When a pin is dedicated to an internal peripheral function the associated DDR bit in some cases still controls whether the line is an input or an output. The SPI, timer input captures and pulse accumulator follow this rule.

Several functions override the state of the DDRs. The SCI forces the I/O state of the two associated Port D lines. The timer also forces the I/O state of each associated Port A line when an output compare using a Port A or Port J line is enabled. Both the PWM timer and the event counter also override the associated DDR bits. In these cases the data direction bits will have no effect on these lines.

When a pin is dedicated to an on-chip peripheral function, writes to the associated PORTx bit do not affect the pin but are stored in an internal latch such that, if the pin becomes available for general purpose output, the driven level will be the last value written to the PORTx bit.

4.1 MODE DEPENDENCIES

The MCU mode is controlled by two mode select bits (SMOD and MDA) in the HPRIO register which are in turn controlled by the two mode select pins (MODA and MODB).

In the expanded modes (normal expanded and test), ports B, C, E and F, R/$\overline{W}$, and $\overline{LIR}$ are dedicated to address, data, and control signals and may not be used for alternate I/O functions. In order to

preserve the Port functions that are displaced by the expanded modes, these functions become externally accessible functions so that they may be emulated with external hardware, if required. The internal register addresses that become external accesses are PORTB, DDRC, PORTC and PORTF.

4.2 GENERAL PURPOSE I/O (PORTS A, C, D, G, H, AND J)

As general purpose I/O lines, each pin has associated with it a bit in a PORTx data register and a bit in the corresponding position in a DDRx register. The data direction register (DDRx) is used to specify the primary direction of data on the I/O pin. However, specification of a line as an output does not prevent reading of the line as an input. When a bit configured as an output is read, there are two kinds of data which may be returned, depending on the internal circuitry of the port. Reading Port A, Port H, or Port J returns the values sensed at the pins. Reading Port C, Port D, or Port G returns the values at the inputs to the pin drivers.

When a line is configured as an input by clearing the DDRx bit, the pin becomes a high impedance input. If a write is executed to a line that is configured as an input, the value will not affect the I/O pin but the bit will be stored in an internal latch so that, if the line is later reconfigured as an output, this value will appear at the I/O pin. This operation can be used to preset a value for an output port prior to configuring it as an output, thereby avoiding glitches on the outputs which may be detrimental to the operation of the external system.

Ports C, D, and G each have a wired-OR mode of operation which is controlled by the CWOM, DWOM, and GWOM bits, respectively. If the corresponding xWOM bit is set, the p-channel drivers in the output buffers are disabled.

Note: bits 6 and 7 of Port D and bits 4 through 7 of Port J are not implemented.

4.3 FIXED DIRECTION I/O (PORTS B, E, AND F)

The pins for ports B, E, and F have fixed data directions and consequently do not have data direction registers associated with them. When ports B and F are being used for general purpose I/O, they are configured as output only ports and reading them returns the levels sensed at the inputs of the pin drivers. Port E supports the eight A/D channel inputs but these pins may also be used as general purpose digital inputs. Writing to the Port E address has no meaning or effect.

4.4 PORT A

Port A is an 8-bit bidirectional port. Port A pins can be used as general purpose I/O or for timer functions. Each pin behaves as a general purpose I/O bit by default, unless a timer function using that bit is specifically enabled. As general purpose I/O, the data direction of Port A pins are determined by the corresponding DDRA bits. The directions of Port A bits 0, 1, and 2 are always

controlled by the DDRA bits whether IC3, IC2, and IC1 are enabled or not. Port A bits 3 – 7 are controlled by the DDRA bits only when the associated output compare functions are disabled. Enabling an output compare function forces the corresponding port bit to be an output, irrespective of the state of the DDRA bit. Using any pins of Port A for timer functions has no effect on the ability to read these pins as inputs (because input sensing logic is always connected to the port pins).

The OC2, OC3, OC4, and OC5 lines out of Port A are enabled by pairs of control bits in the TCTL1 register. The output compare 1 function is unique in that it allows automatic timer control of any combination of the five most significant bits of Port A regardless of whether or not they are being used for another timer function. The OC1 function gains control of Port A bits by setting the corresponding bits in the OC1M control register. The IC1, IC2, IC3, and IC4 input of Port A are enabled by pairs of control bits in the TCTL2 register. (See **SECTION 6: PROGRAMMABLE TIMER, REAL TIME INTERRUPT AND PULSE ACCUMULATOR**.)

The IC4 function and OC5 function share the same Port A bit and they are selected by the I4/O5 bit in the PACTL register. The pulse accumulator system is enabled, on Port A bit 7 as an input, by setting the PAEN bit in the PACTL register. Even while the pulse accumulator is enabled, Port A bit 7 may be configured as an output controlled by OC1 or as a general purpose output.

4.4.1 Data Register (PORTA)

	7	6	5	4	3	2	1	0	
$1000	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0	PORTA
RESET:	0	0	0	0	0	0	0	0	
Alternate Pin Function:	PA1	OC2	OC3	OC4	OC5/IC4	IC1	IC2	IC3	
and/or:	OC1	OC1	OC1	OC1	OC1	—	—	—	

READ: Any time (inputs return pin levels; outputs return pin driver input levels).

WRITE: Data stored in an internal latch. (Drives pins only if configured as outputs.)

RESET: General purpose high impedance inputs ($00).

Note: Writes do NOT change pin state when pin is configured for timer output.

4.4.2 Data Direction Register (DDRA)

	7	6	5	4	3	2	1	0	
$1001	DDA7	DDA6	DDA5	DDA4	DDA3	DDA2	DDA1	DDA0	DDRA
RESET:	0	0	0	0	0	0	0	0	

READ: Any time.

WRITE: Any time.

RESET: $00 (all general purpose I/O configured for input only).

0 – Bits set to zero configure the corresponding I/O pins as inputs.

1 – Bits set to one configure the corresponding I/O pins as outputs.

Note: The timer forces each Port A line associated with an enabled output compare to be an output. In such cases the data direction bits will not be changed but will have no effect on these lines. DDRA will revert to controlling the I/O state of a pin when the associated timer output compare is disabled.

4.5 PORT B

Port B is an 8-bit general purpose output port which also supports the external address bus.

In the expanded modes (normal expanded and test), these pins act as the high order address output pins. During each MCU cycle, bits 8 through 15 of the address are driven out of bits 0 through 7 of Port B. When the NHALT bit in the OPT2 register is cleared and the $\overline{\text{HALT}}$ input is pulled low, all output buffers of Port B bits go tri-state.

In the single chip modes (normal and bootstrap), the Port B pins are general purpose output only pins. Reading Port B in these modes returns the sensed levels at the inputs to the Port B pin drivers. The Port B data register is cleared at reset and all Port B bits output logic zeros.

4.5.1 Data Register (PORTB)

	7	6	5	4	3	2	1	0	
$1004	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0	PORTB
RESET:	0	0	0	0	0	0	0	0	
Alternate Pin Function:	A15	A14	A13	A12	A11	A10	A9	A8	

READ: Any time (returns levels sensed at inputs of Port B pin drivers).

WRITE: Data stored in internal latch (drives pins only if configured as general purpose outputs).

RESET: In single chip modes, all Port B pins are general purpose output only pins (all zeros). In expanded modes, all Port B pins are high order address signal outputs.

4.6 PORT C

Port C is an 8-bit bidirectional port. Port C pins serve one of two basic functions depending on the MCU mode selected; bidirectional data lines or general purpose I/O pins. In either mode, if the CWOM bit in the OPT2 register is set, the p-channel drivers in the output buffers are disabled (wired-OR mode).

In the expanded modes (normal expanded and test), these pins act as bidirectional data pins. During the CPU read cycle, data on the Port C pins are latched internally on the falling edge of E. During the CPU write cycle, the internal data is driven out of Port C and is valid at the falling edge of E. During an internal address read cycle with the IRV bit in the OPT2 register set, the internal data is also driven out of Port C and is valid at the falling edge of E. If the MCU is in the Halt state, all Port C bits become tri-state.

In the single chip modes (normal and bootstrap), the Port C pins are general purpose I/O pins. Bits 0 – 7 are input or output pins depending on the corresponding bit of DDRC. While a bit is configured as an output, reading the bit returns the sensed level at the input to the Port C pin driver. At reset, all DDRC bits are cleared and all Port C bits are configured as inputs.

4.6.1 Data Register (PORTC)

	7	6	5	4	3	2	1	0	
$1006	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0	PORTC
RESET:	0	0	0	0	0	0	0	0	
Alternate Pin Function:	D7	D6	D5	D4	D3	D2	D1	D0	

READ: Any time (inputs return pin level; outputs return pin driver input level).

WRITE: Data stored in internal latch (drives pins only if configured as outputs)

RESET: In single chip modes all Port C pins are configured as general purpose inputs. In expanded modes, Port C is configured as the data bus.

4.6.2 Data Direction Register (DDRC)

	7	6	5	4	3	2	1	0	
$1007	DDC7	DDC6	DDC5	DDC4	DDC3	DDC2	DDC1	DDC0	DDRC
RESET:	0	0	0	0	0	0	0	0	

READ: Any time.

WRITE: Any time.

RESET: $00 (all general purpose I/O configured for input only).

0 – Bits set to zero configure the corresponding I/O pins as inputs.

1 – Bits set to one configure the corresponding I/O pins as outputs.

4.7 PORT D

Port D is a 6-bit bidirectional port which also supports the SCI and SPI systems. When the SCI or SPI is not needed the associated pins may be used for general purpose I/O.

Port D pins serve several functions depending on the MCU mode and various internal control registers. If the DWOM bit in the SPCR register is set, the p-channel drivers in the output buffers are disabled (wired-OR mode).

Port D bit 0 becomes the Receive Data input (RXD) when the SCI receiver is enabled (RE bit in the SCCR2 register set to one). When the RE bit is clear, Port D bit 0 defaults to being a general purpose I/O pin controlled by DDRD.

Port D bit 1 becomes the Transmit Data output (TXD) when the SCI transmitter is enabled (TE bit in the SCCR2 register set to one). When the TE bit is clear, Port D bit 1 defaults to being a general purpose I/O pin controlled by DDRD. Note that the transmit logic will retain control of Port D bit 1 after TE is cleared until all transmit operations have finished, including completion of transmission of data from the serial shifter, a queued idle, or queued break.

In a test mode, the RCKB test bit in the BAUD register may be set. When RCKB is set, the 16X receiver baud rate clock and the 1X transmitter clock are exclusive-ORed and driven out of the Port D bit 1 pin. The RCKB bit can be written only in the test or bootstrap modes and it overrides any other use of the Port D bit 1 pin.

Bits 2 – 5 of Port D are dedicated to the serial peripheral interface function (SPI) whenever the SPE bit in the SPCR register is set (SPI enabled). Note that Port D bit 5 still responds to DDRD bit 5 such that, if the DDR bit is set, the Port D bit 5 pin is given up by the SPI system to become a general purpose output pin if and only if the SPI is in master mode. When the SPE bit is clear, bits 2 – 5 of Port D default to being general purpose I/O pins controlled by DDRD.

The four SPI interface lines are described in greater detail in **SECTION 8: SERIAL PERIPHERAL INTERFACE.**

4.7.1 Data Register (PORTD)

	7	6	5	4	3	2	1	0	
$1008	0	0	PD5	PD4	PD3	PD2	PD1	PD0	PORTD
RESET:	0	0	0	0	0	0	0	0	
Alternate Pin Function:	—	—	$\overline{SS}$	SCK	MOSI	MISO	TXD	RXD	

READ: Any time (inputs return pin level; outputs return pin driver input level). Bits 6 and 7 always read as zeros.

WRITE: Data stored in internal latch (drives pins only if configured as outputs). Writes to bit 6 and 7 have no meaning or effect.

RESET: Bits 0 – 5 are configured as general purpose inputs.

4.7.2 Data Direction Register (DDRD)

	7	6	5	4	3	2	1	0	
$1009	0	0	DDD5	DDD4	DDD3	DDD2	DDD1	DDD0	DDRD
RESET:	0	0	0	0	0	0	0	0	

READ: Any time (reads of bits 6 and 7 always return zeros).

WRITE: Any time (writes to bits 6 and 7 have no meaning or effect).

RESET: $00 (all general purpose I/O configured for input only).

0 – Bits set to zero configure the corresponding I/O pins as inputs.

1 – Bits set to one configure the corresponding I/O pins as outputs.

Bit 5 of Port D is dedicated as the active low slave select ($\overline{SS}$) input, when the SPI system is enabled. In SPI slave mode, DDRD bit 5 has no meaning or effect. In SPI master mode, DDRD bit 5 determines whether Port D bit 5 is an error detect input to the SPI (DDRD bit clear) or a general purpose output line (DDRD bit set).

For bits 2, 3, & 4 (MISO, MOSI, & SCK): if the SPI is enabled and expects the bit to be an input, it will be an input regardless of the state of the DDRD bit; if the SPI is enabled and expects the bit to be an output, it will be an output **only** if the DDRD bit is set.

4.8 PORT E

Port E is an 8-bit port which supports the 8-channel A/D converter. Any channels not used for A/D may be used as general purpose inputs.

Since there is no output drive logic associated with Port E there is no DDRE register. Port E pins are capable of withstanding negative 1 volt transients (current limited by an external 10 kΩ or greater resistor) without affecting MCU operation, other than possible loss of the A/D input sample for a conversion being performed at the time of the transient. Using Port E pins as A/D inputs does not affect the ability to read these pins as static inputs. However, reading Port E during an A/D conversion sequence may inject noise into the analog input signals and may result in reduced accuracy of the A/D result. Note that performing a digital read of Port E, with levels other than V_{DD} or V_{SS} on the Port E pins, will result in greater power dissipation during the read cycle.

4.8.1 Data Register (PORTE)

	7	6	5	4	3	2	1	0	
$1**0**0A	PE7	PE6	PE5	PE4	PE3	PE2	PE1	PE0	PORTE
RESET:	U	U	U	U	U	U	U	U	
Alternate Pin Function:	AN7	AN6	AN5	AN4	AN3	AN2	AN1	AN0	

READ: Any time (returns sensed levels at Port E pins).

WRITE: Has no meaning or effect.

RESET: Does not affect this address.

4.9 PORT F

Port F is an 8-bit general purpose output port which also supports the external address bus.

In the expanded modes (normal expanded and test), these pins act as the low order address output pins. During each MCU cycle, bits 0 – 7 of the address are driven out of bits 0 – 7 of Port F. When the NHALT bit in the OPT2 register is cleared and the HALT input is pulled low, all output buffers on Port F pins go tri-state.

In the single chip modes (normal and bootstrap), the Port F pins are general purpose output only pins. Reading Port F in these modes returns the sensed levels at the inputs to the Port F pin drivers. The Port F data register is cleared at reset and all Port F bits output logic zeros.

4.9.1 Data Register (PORTF)

	7	6	5	4	3	2	1	0	
$1005	PF7	PF6	PF5	PF4	PF3	PF2	PF1	PF0	PORTF
RESET:	0	0	0	0	0	0	0	0	

READ: Any time (returns sensed levels at inputs of Port F pin drivers).

WRITE: Data stored in internal latch (drives pins only if configured as general purpose outputs).

RESET: In single chip modes, all Port F pins are general purpose output only pins (all zeros). In expanded modes, all Port F pins are low order address signal outputs.

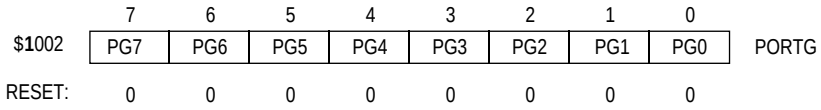
4.10 PORT G

Port G is an 8-bit bidirectional port. Port G pins serve one of two basic functions depending on the MCU mode selected; bidirectional data lines or general purpose I/O pins. In either mode, if the GWOM bit in the OPT2 register is set, the p-channel drivers in the output buffers are disabled (wired-OR mode).

Port G may be programmed as an input or an output under software control. The data direction of each pin is determined by the state of the corresponding bit in the Port G data direction register (DDRG). A pin is configured as an output if its corresponding DDRG bit is set to a logic one; a pin is configured as an input if its corresponding DDRG bit is cleared. At reset, all DDRG bits are cleared, which configure all Port G pins as inputs.

During the programmed output state, a read of Port G actually reads the value of output data latch and not the pin level. When the GWOM bit in OPT2 is set, the p-channel drivers of output buffers are disabled (wired-OR mode).

4.10.1 Data Register (PORTG)



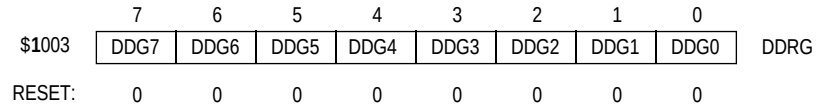
	7	6	5	4	3	2	1	0	
$1002	PG7	PG6	PG5	PG4	PG3	PG2	PG1	PG0	PORTG
RESET:	0	0	0	0	0	0	0	0	

READ: Any time (inputs return pin levels; outputs return pin driver input levels).

WRITE: Data stored in an internal latch (drives pins only if configured as outputs).

RESET: General purpose high impedance inputs ($00)

4.10.2 Data Direction Register (DDRG)



	7	6	5	4	3	2	1	0	
$1003	DDG7	DDG6	DDG5	DDG4	DDG3	DDG2	DDG1	DDG0	DDRG
RESET:	0	0	0	0	0	0	0	0	

READ: Any time.

WRITE: Any time.

RESET: $00 (all general purpose I/O configured for input only).

0 – Bits set to zero configure the corresponding I/O pins as inputs.

1 – Bits set to one configure the corresponding I/O pins as outputs.

4.11 PORT H

Port H is an 8-bit general purpose I/O port. Three of the pins are used for the event counter (see **SECTION 11: EVENT COUNTER**). Four of the remaining pins are used for the 4-channel PWM timer (see **SECTION 10: PULSE WIDTH MODULATION TIMER**). The remaining pin is used for the memory ready input signal (MRDY). Any pins not used for these functions may be used as general purpose I/O.

When the PWENx bit in the PWEN register is set, the associated Port H bit is forced to be a PWM output regardless of the state of the DDRH bit. This does not change the state of the DDRH bit. When the PWENx bit in the PWEN register is cleared, the pin defaults to being a general purpose I/O. The data direction of the pin is determined by the state of the corresponding DDRH bit.

Bits 4 and 5 of Port H are used as the EVI2 and EVI1 inputs or as general purpose I/O. The data direction of Port H bits 4 and 5 is always under the control of DDRH bits 4 and 5, whether EVI2 and EVI1 are enabled or not. If the pin is configured as an output by its DDRH bit, the output data of the Port H register is also the input data of the event counter.

Bit 6 of Port H is used as the event output (EVO) or as general purpose I/O. When the EVOEN bit in the EVCTL register is set, bit 6 of Port H becomes EVO regardless of the state of DDRH bit 6. This does not change the state of DDRH bit 6. When the EVOEN bit in the EVCTL register is cleared, the data direction of the pin is under the control of DDRH bit 6.

In the expanded non-multiplexed and test modes, bit 7 of Port H is used as the memory ready signal (MRDY) or as general purpose I/O. When the MRDY bit in the OPT2 register is set, bit 7 of Port H becomes the memory ready input regardless of the state of DDRH bit 7. When the MRDY bit in the OPT2 register is cleared, the data direction of the pin is under the control of DDRH bit 7. In the single chip and bootstrap modes, bit 7 of Port H is a general purpose I/O pin and the data direction of the pin is determined by the state of DDRH bit 7.

Reading Port H reads the levels sensed at the pins regardless of the DDRH, PWENx, and EVOEN bits. All DDRH bits are cleared at reset.

4.11.1 Data Register (PORTH)

	7	6	5	4	3	2	1	0	
$1033	PH7	PH6	PH5	PH4	PH3	PH2	PH1	PH0	PORTH
RESET:	0	0	0	0	0	0	0	0	
Alternate Pin Function:	MRDY	EVO	EVI1	EVI2	PW4	PW3	PW2	PW1	

READ: Any time (inputs return pin levels, outputs return pin driver input levels).

WRITE: Data stored in an internal latch (drives pins only if configured for output).

RESET: General purpose high impedance inputs ($00).

4.11.2 Data Direction Register (DDRH)

	7	6	5	4	3	2	1	0	
$1034	DDH7	DDH6	DDH5	DDH4	DDH3	DDH2	DDH1	DDH0	DDRH
RESET:	0	0	0	0	0	0	0	0	

READ: Any time

WRITE: Any time

RESET: $00 (all general purpose I/O configured for input only)

0 – Bits set to zero configure the corresponding I/O pins as inputs.

1 – Bits set to one configure the corresponding I/O pins as outputs.

Note: The pulse width modulation timer forces the I/O state to be an output for each Port H line associated with an enabled PWM. In such cases, the data direction bits will not be changed but have no effect on these lines. DDRH will revert to controlling the I/O state of a pin when the associated function is disabled. The event counter does not force the state of any of the associated pins.

4.12 PORT J

Port J is a general purpose 4-bit I/O port which also supports some of the timer functions (see **SECTION 6: PROGRAMMABLE TIMER, REAL TIME INTERRUPT AND PULSE ACCUMULATOR**). Two of the pins are used for input capture/output compare. One of the pins is used as a clock input for the timer counter registers. The one remaining pin is for general purpose I/O. Any channels not used for timers may be used as general purpose I/O.

Bit 0 of Port J is used as the TCK input or for general purpose I/O. The data direction is always under the control of DDRJ bit 0, whether TCK is enabled or not. If the pin is configured as an output by DDRJ bit 0, the output data in Port J bit 0 becomes the input data to TCK.

Bits 1 and 2 of Port J is used for output compare or input capture or as general purpose I/O. If the associated output compare is not enabled, the data directions of these pins are dependent on DDRJ bits 1 and 2. If an output compare is enabled, the associated pin is configured automatically as an output. If a pin is programmed as an output and input capture is enabled, the output data becomes the input data to the input capture.

Bit 3 of Port J is a general purpose I/O pin and the data direction of the pin is determined by the state of DDRJ bit 3. All DDRJ bits are cleared at reset.

4.12.1 Data Register (PORTJ)

	7	6	5	4	3	2	1	0	
$1031	0	0	0	0	PJ3	PJ2	PJ1	PJ0	PORTJ
RESET:	0	0	0	0	0	0	0	0	
Alternate Pin Function:	—	—	—	—	—	O7/I6	O6/I5	TCK	

READ: Any time (inputs return pin levels, outputs return pin driver input levels).

WRITE: Data stored in an internal latch (drives pins only if configured as outputs).

RESET: General purpose high impedance inputs ($00).

4.12.2 Data Direction Register (DDRJ)

	7	6	5	4	3	2	1	0	
$1032	0	0	0	0	DDJ3	DDJ2	DDJ1	DDJ0	DDRJ
RESET:	0	0	0	0	0	0	0	0	

READ: Any time.

WRITE: Any time.

RESET: $00 (all general purpose I/O configured for input only).

0 – Bits set to zero configure the corresponding I/O pins as inputs.

1 – Bits set to one configure the corresponding I/O pins as outputs.

Note: The timer forces the I/O state to be an output if output compare 6 is enabled on Port J bit 1 (OM6 or OL6 is a 1 and I5/O6 is a 0). The timer also forces the I/O state to be an output if output compare 7 is enabled on Port J bit 2 (OM7 or OL7 is a 1 and I6/O7 is a 0). In this case, the data direction bit is not changed but has no effect on this line. The DDRJ bit will revert to controlling the I/O state of the pin when timer output compare 6 is disabled. TCK does **not** force the state of the line associated with it.

4.13 EXPANDED BUS (PORTS B, C, F)

The MC68HC11G5 has a non-multiplexed expansion bus. This simplifies system design, as demultiplexing circuitry is not required. It also eliminates the need for an address strobe line.

The user gains three additional ports when the part is used in single chip mode. Port B provides the high order addresses in expanded mode, but may be used as a general purpose output port in single chip mode. Similarly, Port F provides the low order addresses, but may be used as a general purpose output port in single chip mode. Port C is the data bus in expanded mode, but may be used as general purpose I/O in single chip mode. Port C has a data direction register for use in single chip mode.

In order to allow emulation of all MC68HC11G5 functions, even in the expanded modes, the functions displaced by specification of the expanded mode become externally addressed functions. The registers which become external accesses are PORTC, DDRC, PORTB and PORTF.

4.13.1 R/$\overline{W}$

The read/write output signal (R/$\overline{W}$) is a dedicated function when the MC68HC11G5 is in normal expanded mode or test mode. The timing of this signal is the same as the timing for a Port B address output, except for the hold time from the falling edge of E, which is extended so that no special circuitry is needed in a multi-board expanded system. This output reflects the state of the internal CPU R/$\overline{W}$ signal.

In the single chip and bootstrap modes, the R/$\overline{W}$ pin is a dedicated output which is always high (read). This permits changing from one of these modes to an expanded mode, without the risk that the R/$\overline{W}$ line might have been in the “write” state, causing bus conflicts or inadvertent external memory changes.

4.13.2 Memory Ready (MRDY)

A memory ready function is available on the MC68HC11G5. This allows interfacing to slow peripherals and dual ported RAM, and to dynamic RAM without a hidden refresh.

When the memory ready function is enabled, MRDY is used to stretch the CPU timing and E-clock to allow a longer access time. Note that internal clocks to timers and baud rate generators continue to run at the normal rate so that the timer and serial systems are not affected.

Port H bit 7 is used as the memory ready line (MRDY). When this line is low, an external access will be stretched until the line goes high. During the stretch time the address lines will be held and E will be kept in the high state. For write operations the data will be held and for read operations the data bus lines will be inputs. When MRDY goes high, E will fall thus ending the cycle, and will continue at a normal rate until the next external access during which MRDY is low. Note that the bus will only stretch for integral numbers of E-clock cycles.

4.13.3 Options Register 2 (OPT2)

	7	6	5	4	3	2	1	0	
$1038	GWOM	CWOM	0	IRV	0	0	MRDY	NHALT	OPT2
RESET:	0	0	0	—	0	0	0	1	

GWOM — PORT G Wired-Or Mode

READ: Any time.

WRITE: Any time.

0 – Port G operates normally.

1 – Port G outputs are open drain (used to facilitate testing).

CWOM — PORT C Wired-Or Mode

READ: Any time.

WRITE: Any time.

0 – Port C operates normally.

1 – Port C outputs are open drain (used to facilitate testing).

IRV — Internal Read Visibility

READ: Any time.

WRITE: If SMOD=1, any time. If SMOD=0, only one write is allowed.

RESET: Test and bootstrap modes – 1; single chip and expanded modes – 0.

0 – No visibility of internal reads on external bus.

1 – Data from internal reads is driven out on the external data bus.

MRDY — Memory Ready Enable

READ: Any time.

WRITE: Any time.

0 – Memory Ready is disabled, Port H bit 7 is general purpose I/O.

1 – Memory Ready is enabled, Port H bit 7 is forced to be an input used as the MRDY line. External bus accesses will be stretched as long as this line is held low.

NHALT — Enable Halt Function

READ: Any time

WRITE: Any time

0 – Halt enabled (a logic low level detected on the $\overline{\text{HALT}}$ pin will cause the part to go into the HALT state at the completion of the present instruction).

1 – Halt disabled.

SECTION 5

RESETS, INTERRUPTS AND LOW POWER MODES

This section describes the internal and external resets and interrupts of the MC68HC11G5 and its two low power consumption modes.

5.1 RESETS

The MCU can be reset in four ways:

1. An active-low input to the $\overline{\text{RESET}}$ pin
2. A power-on-reset function
3. A clock monitor failure
4. A computer operating properly (COP) watchdog timer timeout

The $\overline{\text{RESET}}$ input circuitry includes a Schmitt trigger which senses the $\overline{\text{RESET}}$ line logic level.

5.1.1 $\overline{\text{RESET}}$ Pin

The $\overline{\text{RESET}}$ pin is used to reset the MCU and allow an orderly software startup procedure. When a reset condition is sensed, this pin is driven low by an internal device for four E-clock cycles, then released, and two E-clock cycles later it is sampled. If the pin is still low, it means that an external reset has occurred. If the pin is high, it implies that the reset was initiated internally by either the watchdog timer (COP) or the clock monitor. This method of differentiation between internal and external reset conditions assumes that the reset pin will rise to a logic one in less than two E-clock cycles once it is released, and that an externally generated reset should stay active for at least eight E-clock cycles.

5.1.2 Power-on-reset (POR)

Power-on-reset occurs when a positive transition is detected on V_{DD}. This reset is used strictly for power turn-on conditions and should not be used to detect any drop in the power supply voltage. If the external $\overline{\text{RESET}}$ pin is low at the end of the power-on-reset delay time, the processor remains in the reset state until $\overline{\text{RESET}}$ goes high.

5.1.3 Computer Operating Properly (COP) Reset

The MCU contains a watchdog timer which automatically times out unless it is reset within a specific time by a program reset sequence. If the COP watchdog timer is allowed to timeout, a reset is generated which drives the RESET pin low to reset the MCU and the external system.

The COP reset function can be enabled by programming the NOCOP control bit in the system configuration register (CONFIG). Once programmed, this control bit remains set (or cleared), even when no power is applied, and the COP function is enabled or disabled independent of resident software. Protected control bits CR0 and CR1 in the configuration options register (OPTION) allow the user to select one of four COP timeout periods. Table 5-1 shows the relationship between CR0 and CR1 and the timeout period for various system clock frequencies.

Table 5-1. COP Timeout Periods

CR1	CR0	E ÷ 2^{15} Divided By	XTAL = 2^{23} Timeout – 0 / +15.6 ms	XTAL = 8.0 MHz Timeout – 0 / +16.4 ms	XTAL = 4.9152 MHz Timeout – 0 / +26.7 ms	XTAL = 4.0 MHz Timeout – 0 / +32.8 ms	XTAL = 3.6864 MHz Timeout – 0 / +35.6 ms
0	0	1	15.625 ms	16.384 ms	26.667 ms	32.768 ms	35.556 ms
0	1	4	62.5 ms	65.536 ms	106.67 ms	131.07 ms	142.22 ms
1	0	16	250 ms	262.14 ms	426.67 ms	524.29 ms	568.89 ms
1	1	64	1 s	1.049 s	1.707 s	2.1 s	2.276 s
		E =	2.1 MHz	2.0 MHz	1.2288 MHz	1.0 MHz	921.6 kHz

The sequence for resetting the watchdog timer is as follows:

1) Write $55 to the COP reset register (COPRST)
2) Write $AA to the COP reset register (COPRST)

Both writes must occur in this sequence prior to the timeout, but any number of instructions can be executed between the two writes.

5.1.4 Clock Monitor Reset

The MCU contains a clock monitor circuit which measures the E-clock frequency. The clock monitor function is enabled by the CME control bit in the OPTION register. Upon detection of a slow or absent clock, the clock monitor circuit (if enabled by CME = 1) will cause a system reset to be generated. If the E-clock input rate is above approximately 200 kHz, then the clock monitor does not generate a reset. If the E-clock is lost or its frequency falls below 10 kHz, then a reset is generated, and the RESET pin is driven low to reset the external system.

Special considerations are needed when using STOP and the clock monitor in the same system. Since the STOP function causes the clocks to be halted, the clock monitor function will generate a reset sequence if it is enabled prior to the STOP mode being entered. For systems which do not expect or want a STOP function, this interaction can be useful to detect the unauthorized execution of a STOP instruction which could not be detected by the COP watchdog system. On the other hand, in systems which utilise both the STOP and clock monitor functions, this interaction means that the CME bit must be written to zero just prior to executing a STOP instruction and should be written back to one as soon as the MCU resumes execution.

5.1.5 Configuration Options Register (OPTION)

The bits in this register control certain configuration options, most of which can be changed only during the first 64 cycles after reset in normal operating modes.

	7	6	5	4	3	2	1	0	
$1039	ADPU	CSEL	IRQE	DLY	CME	0	CR1	CR0	OPTION
RESET:	0	0	0	1	0	0	0	0	

READ: Any time.

WRITE: Bits 3, 6, and 7 may be written at any time.
Bits 0, 1, 4, and 5 may be written once only in the normal operating modes, and only during the first 64 cycles after reset. After this time the bits are read-only in the normal operating modes (SMOD = 0). In the special test and bootstrap modes (SMOD = 1), writes are always permitted.

RESET: $10.

ADPU — A/D Power Up

0 – A/D system powered down to save supply current.

1 – A/D system powered up (allow about 100 μs for stabilization).

CSEL — Clock Select

This bit should be set to one if the E-clock is less than 1 MHz.

0 – A/D uses the system E-clock (must be 1.0 MHz or greater).

1 – A/D uses an internal R-C clock source (about 1.5 MHz).

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

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

1 – $\overline{\text{IRQ}}$ configured to respond only to falling edges.

DLY — Enable Oscillator Start-up delay on exit from STOP

0 – No stabilization delay imposed on exit from STOP mode.

1 – A stabilization delay is imposed before processing resumes after STOP.

This bit is set during reset and controls whether or not a relatively long stabilization delay is imposed before processing can resume after a STOP period. If an external clock signal

is supplied, this delay can be inhibited so that processing can be resumed within a few cycles of a wake-up from STOP mode. When DLY is set, a 4064 E-clock cycle delay is imposed to allow oscillator stabilization.

CME — Clock Monitor Enable

0 – Clock monitor disabled; a slow clock may be used.

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

In order to use both STOP and the clock monitor, the CME bit should be written to zero prior to executing a STOP instruction and rewritten to one after recovery from STOP.

CR1 and CR0 — COP Timer Rate select bits

The COP system is driven by a constant frequency of E divided by 215. CR1 and CR0 specify an additional divide-by factor to arrive at the COP timeout rate (see Table 5-1).

5.1.6 State After Reset

The MCU mode is a function of the state of the two mode select pins. Enable bits for the ROM and the COP system are contained in the CONFIG register, which is initialized during reset, and are dependent on the mode which has been selected. In the single chip, bootstrap and test modes, the ROM is on; in expanded mode the ROM is off. In the single chip and expanded modes the COP is on; in bootstrap and test modes the COP is off.

The remainder of the system configuration is controlled via registers. Most of these registers and control bits are forced to a specific state during reset. If a user requires a different configuration, he must write different information into these registers.

Most of the configuration state after reset is independent of the MCU mode selected. Those features which are specifically dependent on the mode are described in the following paragraphs.

CPU — The CPU fetches the restart vector from $FFFE, $FFFF ($BFFE, $BFFF in bootstrap mode or test mode) during the first two cycles after reset and begins executing instructions. The stack pointer and other CPU registers are indeterminate immediately after reset, except for the X and I interrupt mask bits in the CCR, which are set so that interrupt requests will be masked, and the S bit, also in the CCR, which is set so that the STOP instruction is disabled.

Memory Map — Immediately after reset the internal memory map of the MC68HC11G5 has 16 kilobytes of ROM located at the top of memory from $C000 – $FFFF (except in expanded mode where the ROM is disabled and these bytes are external accesses), 512 bytes of RAM located at the bottom of memory ($0000 – $01FF) and 128 bytes of internal register and I/O space located at ($1000 – $107F). If the bootstrap mode is selected, the memory map is the same except for the addition of a small internal bootstrap ROM which is located at $BF00 – $BFFF. When a vector fetch occurs in bootstrap mode, the A14 bit is forced low so that the vectors in the bootstrap ROM are selected. In the test mode, A14 is also forced low during vector fetches but the bootstrap ROM is not enabled so vectors are fetched from external memory located at $BFC0 – $BFFF.

Parallel I/O — If reset in an expanded mode, the pins used by the parallel I/O functions are dedicated to the expansion bus and the parallel I/O functions become externally accessed functions to allow emulation. If reset in single chip mode, the CWOM bit is initialized to zero (Port C not wired-OR

mode). Port C is initialized as an input port (DDRC = $00). Ports B and F are general purpose output ports with all bits initialized to zero. R/$\overline{W}$ outputs a logic high level all the time. Ports A, D, G, H, and J are configured as general purpose high-impedance inputs. Port E pins are configured as inputs.

Timer — The timer system is initialized during reset to a count of $0000. The prescaler bits are cleared (to zero) so that the count rate equals the E-clock rate. All output compare registers are initialized to $FFFF to guarantee the maximum time before any successful compare. All input capture registers are indeterminate after reset. The OC1M register is cleared so successful OC1 compares will not affect any I/O pins. The other output compare functions are configured so as not to affect any I/O pins on successful compares. All input capture edge detector circuits are configured for "capture disabled" operation. The timer overflow interrupt flag and all timer function interrupt flags are cleared and all timer interrupts are disabled by their mask bits being cleared.

Real Time Interrupt — The real time interrupt flag is cleared and automatic hardware interrupts are masked by the RTII bit being clear. The rate control bits RTR1, RTR0 are cleared to zero, thus selecting the shortest real time interrupt period.

Pulse Accumulator — The pulse accumulator system is disabled at reset. The PAIF and PAOVF flags are cleared to indicate no pulse accumulator interrupts pending. The PAII and PAOVI interrupt enable bits are cleared to inhibit pulse accumulator interrupts.

COP — The COP watchdog system is enabled if the NOCOP control bit is a zero and disabled if NOCOP is one. The COP rate is set for the shortest duration timeout. If a different rate is required then it must be initialized within 64 E-clock cycles after reset.

SCI Serial I/O — The reset condition of the SCI system is independent of the operating mode. At reset, the SCI baud rate is indeterminate and must be established by a software write to the BAUD register (note that in bootstrap mode software in the bootstrap ROM initializes the SCI system and the baud rate). The frame format is configured for 8-bit data (M = 0). All transmit and receive interrupts are masked and both the transmitter and receiver are disabled (turned off) so the port pins associated with the SCI default to being general purpose I/O lines. The send break and receiver wake-up functions are disabled. Although the wake-up function is disabled (RWU = 0), the WAKE control bit is initialized to 0, thus selecting the 'idle line' mode of wake-up. The TDRE and TC status bits are both set to one indicating that there is no transmit data in either the transmit data register or the transmit serial shift register. The receiver related status bits RDRF, IDLE, OR, NF, and FE are all cleared by reset.

SPI Serial I/O — The SPI system is disabled during reset. The port pins associated with this function default to being general purpose I/O pins.

A to D — The A/D system is disabled by reset. Both the ADPU and CSEL bits are cleared to zero, disabling the analog circuitry of the A/D and the internal R-C oscillator. The bits in the ADCTL control register are indeterminate following reset. In any case, a write must be performed to this register in order to initiate a conversion sequence.

System — The "Highest Priority I" interrupt defaults to the external interrupt pin $\overline{IRQ}$ by PSEL4 – PSEL0 being set equal to 00110. 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 internal read visibility

control bit (IRV) in the OPT2 register is initialized to zero in normal modes to protect the system from potential bus conflict with the external system. IRV is initialized to one if the MC68HC11G5 is reset in a special mode (SMOD = 1) to enable internal read visibility for test and debug purposes. The $\overline{\text{IRQ}}$ pin is configured for level sensitive operation (for wired-OR systems). The DLY control bit in the OPTION register is set to specify that an oscillator start-up delay will be imposed upon recovery from STOP mode. The clock monitor system is disabled by the CME bit in the OPTION register being set to zero. When the system is reset in a special mode (SMOD = 1), the DISR control bit in the TEST1 register is initialized to logic one so that COP and clock monitor failures will not generate a reset. In the normal modes (SMOD = 0), the DISR control bit is cleared to allow normal operation of the COP and clock monitor systems.

5.2 INTERRUPTS

Excluding reset type interrupts, the MC68HC11G5 has 22 hardware interrupt sources and one software interrupt source. These interrupts can be divided into two categories, maskable and non-maskable. Twenty of the interrupt sources can be masked using the I bit in the condition code register (CCR). All the on-chip hardware interrupts are individually masked by local control bits. The software interrupt (SWI) is non-maskable. The external input to the $\overline{\text{XIRQ}}$ pin is considered a non-maskable interrupt because it cannot be masked by software once it is enabled. However, it is masked during reset and upon receipt of an interrupt signal at the $\overline{\text{XIRQ}}$ pin. Illegal opcode is also a non-maskable interrupt.

5.2.1 Interrupt Vector Assignments

In all normal operating modes the interrupt vectors are located at the top of the address space ($FFC0 through $FFFF). In the bootstrap mode the interrupt vectors are located at $BFC0 through $BFFF so that they exist in the internal bootstrap ROM. In the special test mode the interrupt vectors also reside at $BFC0 – $BFFF but the internal bootstrap ROM is disabled so vectors are fetched from external memory.

Table 5-2 provides a list of the interrupts with a vector location in memory for each, as well as the condition code register and control bits that mask each interrupt. Figure 5-1 shows the interrupt stacking order.

Table 5-2. Interrupt Vector Masks and Assignments

Vector Address	Interrupt Source	Masked by	Priority (1= High)
FFC0, FFC1	Reserved	—	—
*	*		
*	*		
FFCA, FFCB	Reserved	—	—
FFCC, FFCD	Event 2	I Bit	22
FFCE, FFCF	Event 1	I Bit	21
FFD0, FFD1	Timer Overflow 2	I Bit	18
FFD2, FFD3	Timer OC7/IC6	I Bit	16
FFD4, FFD5	Timer OC6/IC5	I Bit	15
FFD6, FFD7	SCI Serial System	I Bit	24
FFD8, FFD9	SPI Serial Transfer Complete	I Bit	23
FFDA, FFDB	Pulse Accumulator Input Edge	I Bit	20
FFDC, FFDD	Pulse Accumulator Overflow	I Bit	19
FFDE, FFDF	Timer Overflow 1	I Bit	17
FFE0, FFE1	Timer OC5/IC4	I Bit	14
FFE2, FFE3	Timer Output Compare 4	I Bit	13
FFE4, FFE5	Timer Output Compare 3	I Bit	12
FFE6, FFE7	Timer Output Compare 2	I Bit	11
FFE8, FFE9	Timer Output Compare 1	I Bit	10
FFEA, FFEB	Timer Input Capture 3	I Bit	9
FFEC, FFED	Timer Input Capture 2	I Bit	8
FFEE, FFEF	Timer Input Capture 1	I Bit	7
FFFO, FFF1	Real Time Interrupt	I Bit	6
FFF2, FFF3	$\overline{\text{IRQ}}$ (external pin)	I Bit	5
FFF4, FFF5	$\overline{\text{XIRQ}}$ pin (pseudo NMI)	X Bit	4
FFF6, FFF7	SWI	none	*
FFF8, FFF9	Illegal Opcode Trap	none	*
FFFA, FFFB	COP Failure (Reset)	none	3
FFFC, FFFD	COP Clock Fail Monitor (Reset)	none	2
FFFE, FFFF	$\overline{\text{RESET}}$	none	1

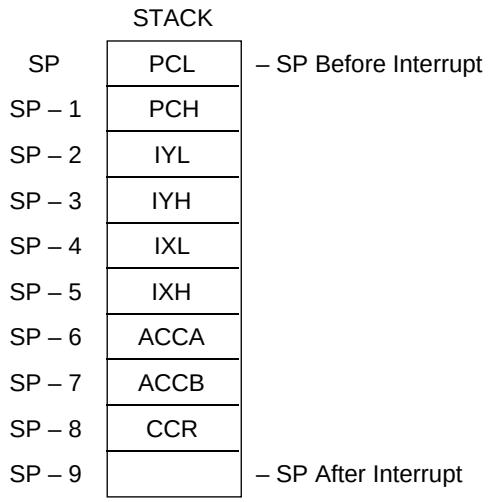


Figure 5-1. Interrupt Stacking Order

5.2.2 Software Interrupt (SWI)

SWI is an instruction rather than a prioritized asynchronous interrupt source. It is non-maskable. In one sense it is lower in priority than any source, because once an interrupt sequence has begun, SWI cannot override it. In another sense, it is higher in priority than any source except reset, because once the SWI opcode is fetched, no other source can be serviced until after the first instruction in the SWI service routine has been executed. SWI causes the I mask bit in the CCR to be set.

5.2.3 Illegal Opcode Trap

Since not all possible opcodes or opcode sequences are defined, an illegal opcode detection circuit has been included in the MC68HC11G5. When an illegal opcode is detected, an interrupt is requested to the illegal opcode vector. It is non-maskable and the illegal opcode vector should never be left uninitialized.

5.2.4 Real Time Interrupt

The real time interrupt (or periodic interrupt) feature on the MC68HC11G5 is configured and controlled by three bits in the PACTL register (RTR2, RTR1 and RTR0 which select one of six interrupt rates based on the MCU E-clock), an interrupt status flag (RTIF) in the TFLG2 register, and a mask bit (RTII) in the TMSK2 register (which enables/inhibits hardware interrupts based on the RTIF flag bit). After reset, one entire real time interrupt period will elapse before the RTIF flag gets set for the first time. (See **SECTION 6: PROGRAMMABLE TIMER, REAL TIME INTERRUPT AND PULSE ACCUMULATOR** for more information on the real time interrupt.)

5.2.5 Interrupt Mask Bits in Condition Code Register

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

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 the I bit are automatically set by hardware after stacking the CCR byte. A return from interrupt (RTI) instruction restores the X and I bits to their pre-interrupt request state.

5.2.6 Priority and Masking Structure

Interrupts obey a fixed hardware priority circuit to resolve simultaneous requests. However, one I bit related interrupt source may be elevated to the highest I bit priority position in the resolution circuit.

Six interrupt sources are not masked by the I bit in the condition codes register and have the fixed priority relationship:

1. Reset
2. Clock failure monitor
3. COP failure
4. Illegal opcode
5. SWI
6. $\overline{\text{XIRQ}}$

SWI is actually an instruction and has highest priority other than reset in the sense that once the SWI opcode is fetched, no other interrupt can be honoured until after the first instruction in the SWI service routine has been executed. The scenario is similar for the illegal opcode.

Each of these sources is an input to the priority resolution circuit. The highest I bit masked priority input to the resolution circuit is assigned under software control (via the HPRIO register) to be connected to any one of the remaining I bit related interrupt sources. In order to avoid timing races the HPRIO register may only be written while the I bit related interrupts are inhibited (I bit in CCR set). An interrupt which is assigned to this high priority position is still subject to masking by any associated control bits or by the I bit in the CCR. The address from which the interrupt vector is fetched is not affected by assigning a source to this higher priority position.

All interrupt sources from on-chip peripherals have software programmable interrupt mask bits which may be used to selectively inhibit automatic hardware response to each interrupt source. In addition the X and I bits in the condition code register act as class inhibit masks to inhibit all sources in the X bit and/or I bit class. Figures 5-2, 5-3 and 5-4 summarize the priority structure and additional mask conditions that lead to the recognition of interrupt requests in the MC68HC11G5.

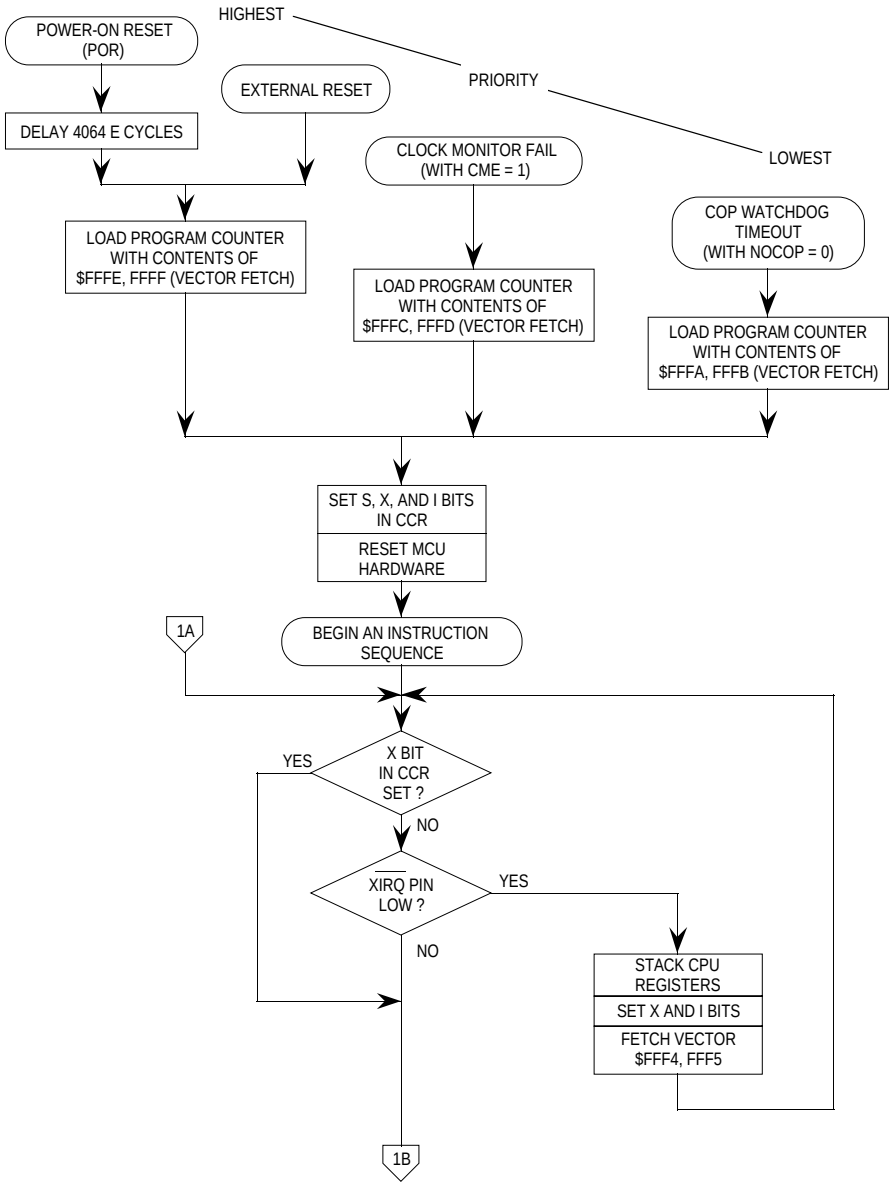


Figure 5-2. Processing Flow out of Resets

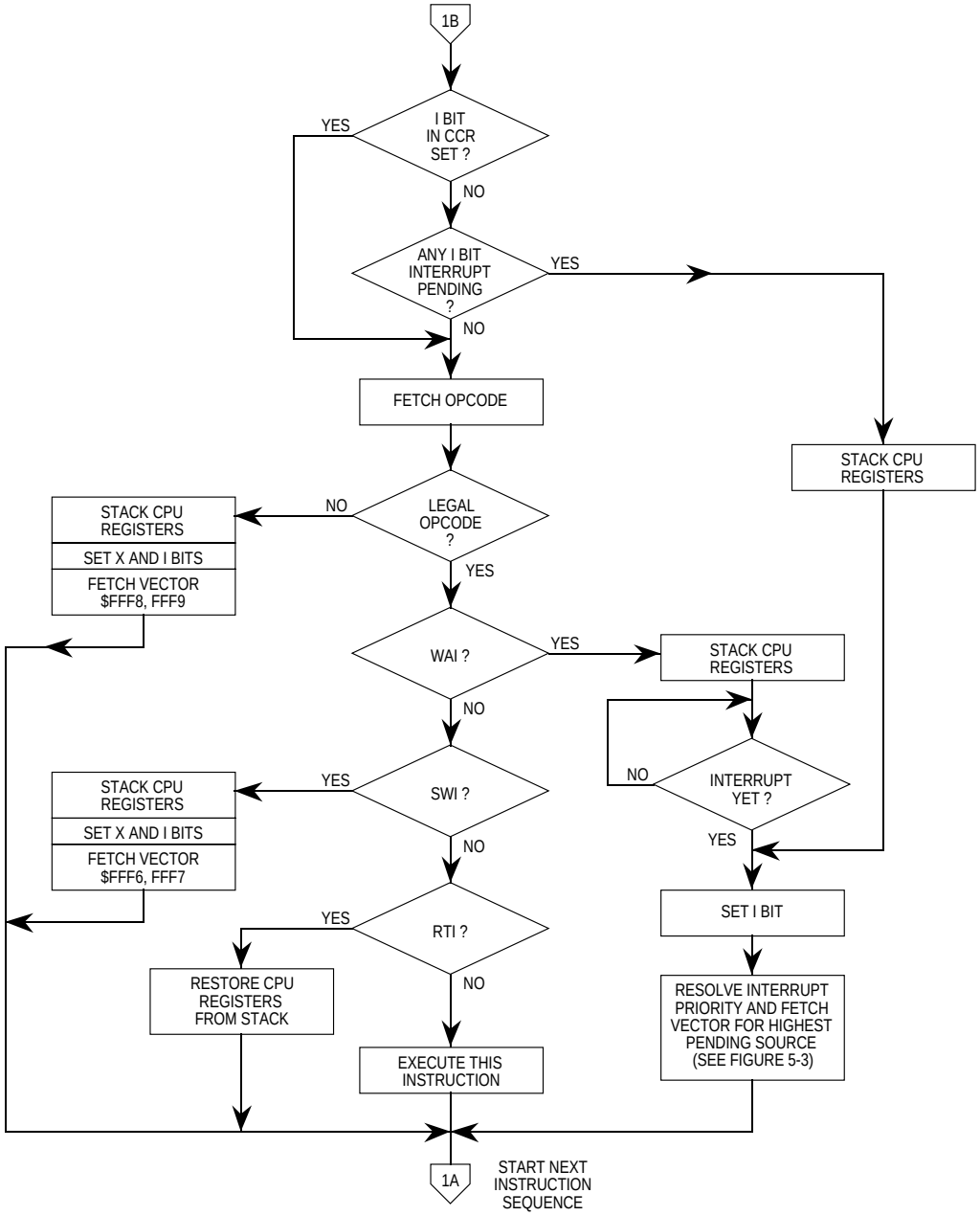


Figure 5-2. Processing Flow out of Resets (contd)

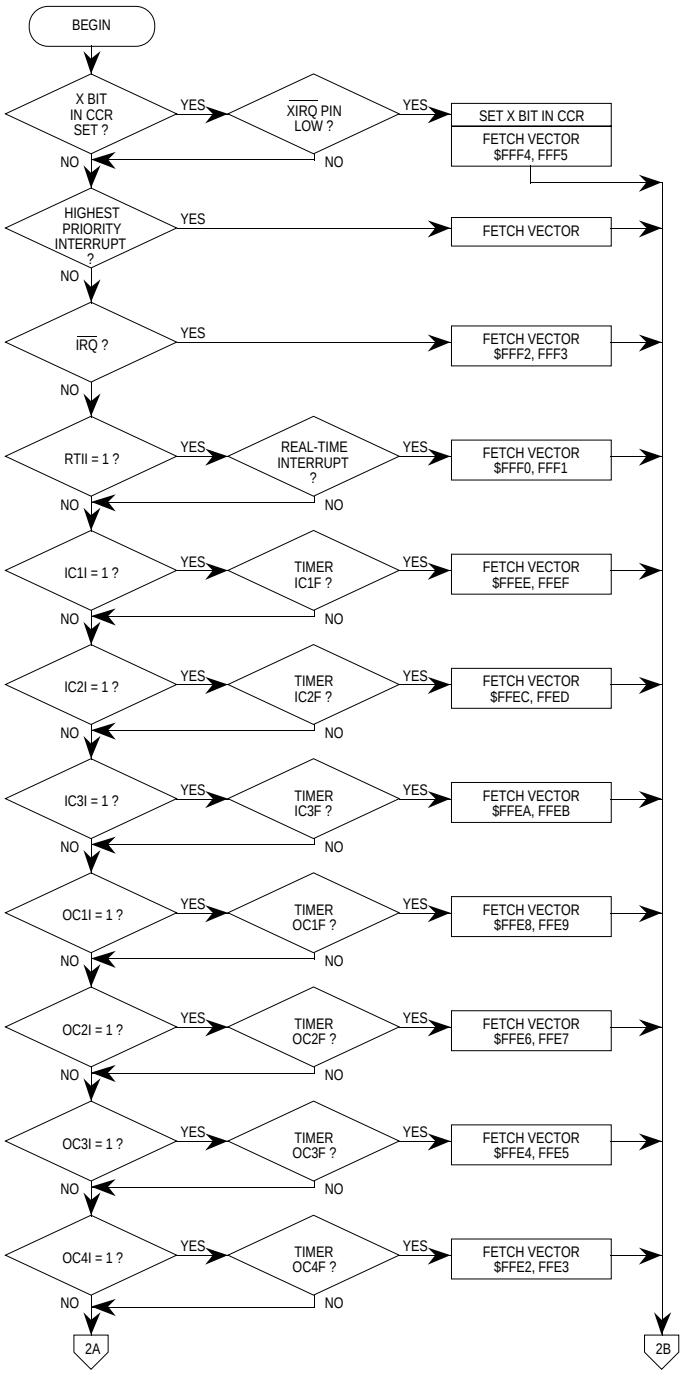


Figure 5-3. Interrupt Priority Resolution

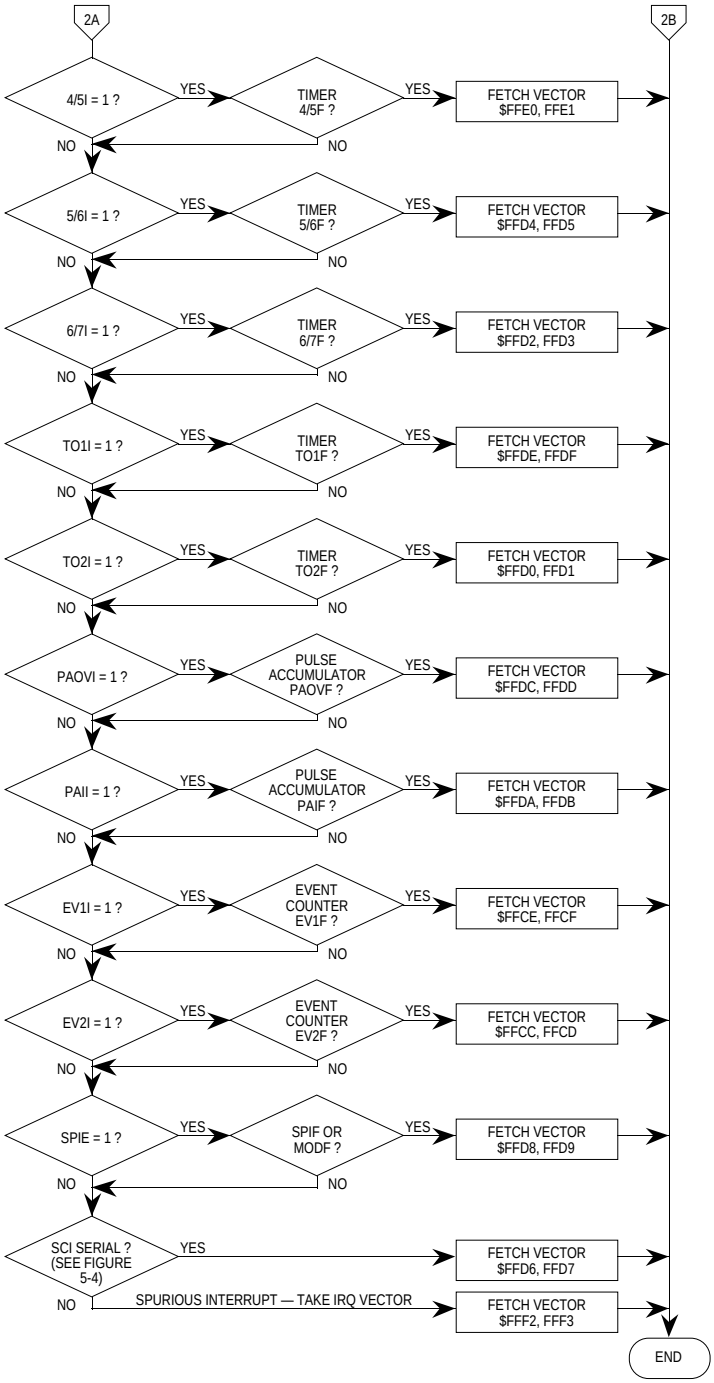


Figure 5-3. Interrupt Priority Resolution (contd)

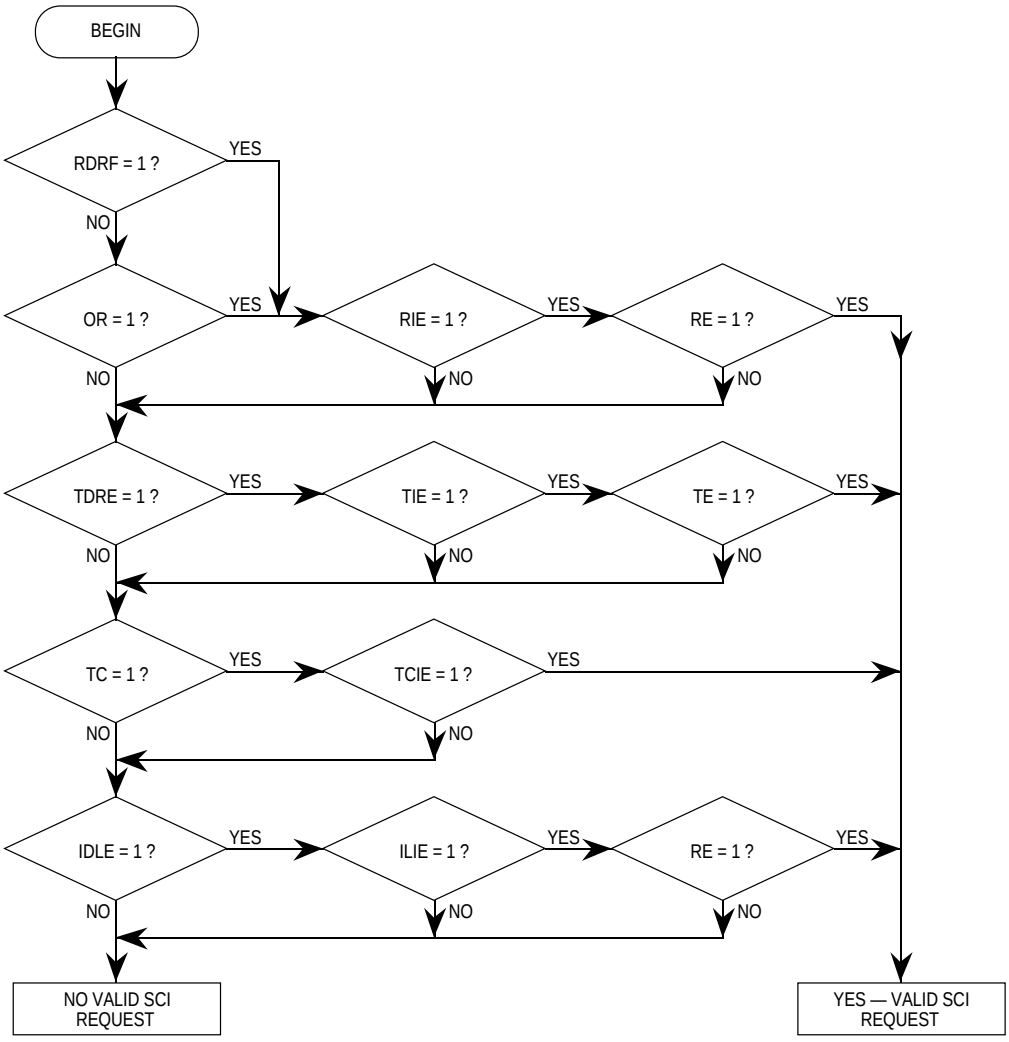


Figure 5-4. Interrupt Source Resolution within SCI

5.2.7 “Highest Priority I” Interrupt and Miscellaneous Register (HPRIO)

	7	6	5	4	3	2	1	0	
$103C	RBOOT	SMOD	MDA	PSEL4	PSEL3	PSEL2	PSEL1	PSEL0	HPRIO
RESET:	—	—	—	0	0	1	1	0	

RBOOT — Read Bootstrap ROM

READ: Any time.

WRITE: Only while SMOD = 1 (test or bootstrap modes).

RESET: Set in Bootstrap mode, cleared in all other modes.

0 – Bootstrap ROM disabled and not in memory map.

1 – Bootstrap ROM enabled in memory map at $BF00 through $BFFF.

SMOD and MDA — Special Mode Select and Mode Select A

READ: Any time.

WRITE: SMOD may only be written to a 0. MDA may be written any time while SMOD = 1, but MDA may only be written once while SMOD = 0.

RESET: These bits reflect the status of the MODB and MODA input pins at the rising edge of reset.

An interpretation of the values of SMOD and MDA is shown in Table 5-3.

Note: SMOD cannot be written to a logic one after being cleared.

Table 5-3. Special Mode Select and Mode Select A Table

Input Pins		Mode Selected	Latched at Reset	
MODA	MODB		SMOD	MDA
0	1	Single Chip	0	0
1	1	Expanded Non-Multiplexed	0	1
0	0	Bootstrap	1	0
1	0	Test	1	1

PSEL4, PSEL3, PSEL2, PSEL1, PSEL0 — Priority Select bits 4 – 0

READ: Any time.

WRITE: Only while I bit in CCR is set (interrupts inhibited).

These five bits are used to specify one I bit related interrupt source which will become the highest priority I bit related source (see Table 5-4).

Table 5-4. Priority Select Bits Table

PSEL4	PSEL3	PSEL2	PSEL1	PSEL0	Interrupt Source Promoted
0	0	0	X	X	Reserved (Default to $\overline{\text{IRQ}}$)
0	0	1	0	0	Reserved (Default to $\overline{\text{IRQ}}$)
0	0	1	0	1	Reserved (Default to $\overline{\text{IRQ}}$)
0	0	1	1	0	$\overline{\text{IRQ}}$ (External pin or Parallel I/O)
0	0	1	1	1	Real Time Interrupt
0	1	0	0	0	Timer Input Capture 1
0	1	0	0	1	Timer Input Capture 2
0	1	0	1	0	Timer Input Capture 3
0	1	0	1	1	Timer Output Compare 1
0	1	1	0	0	Timer Output Compare 2
0	1	1	0	1	Timer Output Compare 3
0	1	1	1	0	Timer Output Compare 4
0	1	1	1	1	Timer OC5/IC4
1	0	0	0	0	Timer Overflow 1
1	0	0	0	1	Pulse Accumulator Overflow
1	0	0	1	0	Pulse Accumulator Input Edge
1	0	0	1	1	SPI Serial Transfer Complete
1	0	1	0	0	SCI Serial System
1	0	1	0	1	Timer OC6/IC5
1	0	1	1	0	Timer OC7/IC6
1	0	1	1	1	Timer Overflow 2
1	1	0	0	0	Event Counter 1
1	1	0	0	1	Event Counter 2
1	1	0	1	0	Reserved (Default to $\overline{\text{IRQ}}$)
1	1	0	1	1	Reserved (Default to $\overline{\text{IRQ}}$)
1	1	1	X	X	Reserved (Default to $\overline{\text{IRQ}}$)

5.3 LOW POWER MODES

The MC68HC11G5 has a WAIT mode to suspend processing and reduce power consumption to an intermediate level. The MC68HC11G5 also offers a STOP mode which turns off all on-chip clocks and reduces power consumption to an absolute minimum while retaining the contents of all 512 bytes of RAM.

5.3.1 WAIT

WAIT mode is invoked by executing the WAI opcode. The CPU registers are stacked and CPU processing is suspended until a qualified interrupt is detected. The interrupt can be an external IRQ, an 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 standby period.

The reduction of power in WAIT mode depends on how many internal clock signals driving on-chip peripheral functions can be shut down. The CPU is always shut down in WAIT mode. The free-running timer system is shut down in WAIT mode if, and only if, the I bit is set to one and the COP system is disabled by NOCOP being set to one. Several other systems may also be in a reduced power consumption state depending upon the state of software controlled configuration control bits. Power consumption by the A/D is not significantly affected by WAIT mode. However, the A/D current can be eliminated by writing the ADPU bit to zero. 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.

The power consumption in WAIT mode is very dependent, therefore, on the particular application.

5.3.2 STOP

STOP mode is invoked by executing the STOP instruction while the S bit in the CCR is equal to zero. If the S bit is not zero then the STOP opcode is treated as a no-op (NOP). STOP mode offers minimum power consumption because all clocks including the crystal oscillator are stopped while in this mode. To exit the STOP mode and resume normal processing, a logic low level must be applied to one of the external interrupts (IRQ or XIRQ) or to the RESET pin. A pending edge-triggered IRQ can also bring the CPU out of STOP mode.

Since all clocks are stopped in this mode, all internal peripheral functions also stop. The data in the internal RAM is retained as long as VDD power is maintained. The CPU state and I/O pin levels are static and are unchanged by the STOP mode. Therefore, when an interrupt comes to restart the system, the MC68HC11G5 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.

In order for the IRQ pin to be used as a means of recovering from the STOP mode, the I bit in the CCR must be clear (IRQ not masked). The XIRQ input may 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 set to zero (XIRQ not masked), the MCU will start up beginning with the stacking

sequence leading to normal service of the $\overline{\text{XIRQ}}$ request. If X is set to one ($\overline{\text{XIRQ}}$ masked or inhibited), then processing will continue with the instruction which immediately follows the STOP instruction, and no $\overline{\text{XIRQ}}$ interrupt service will be requested or pending.

Since the oscillator is stopped in STOP mode, a restart delay may be imposed to allow oscillator stabilization upon leaving the STOP mode. If the internal oscillator is being used, this delay is required; however,if a stable external oscillator is being used, the DLY control bit may be used to by-pass this start-up delay. The DLY control bit is set by reset and may 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 since this will cause DLY to be set again by reset and the restart delay will be imposed. 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.

SECTION 6

PROGRAMMABLE TIMER, REAL TIME INTERRUPT AND PULSE ACCUMULATOR

This section describes the 16-bit programmable timer, the real time interrupt and the pulse accumulator system.

In order to reduce the overhead required to service interrupts, the three timer overflows and the other twelve timer functions each have their own interrupt vector, and each of these interrupts is independently maskable.

6.1 PROGRAMMABLE TIMER

The timer system on the MC68HC11G5 has been derived from the original MC68HC11A8 timer.

There are four fixed output compares (OC), three fixed input captures (IC) and three programmable input captures/output compares (IC/OC).

Secondly, there are two separate “time-of-day” type 16-bit free-running counters, each with its own prescaler. The first counter is associated with the four fixed output compares and two of the programmable input capture/output compares. The second counter is associated with the three fixed input captures and one programmable input capture/output compare.

6.1.1 Counters

The two 16-bit counters (TCNT1 and TCNT2) are clocked by the outputs of separate 3-stage prescalers (divide by 1, 2, 4, or 8) which are in turn driven by the MCU E-clock. (Note that these prescaler values have been changed from the original MC68HC11A8 to include divide-by-2 and to exclude divide-by-16.) The second counter can also be driven by an external clock instead of the internal clock. An external clock cannot be used with counter 1 because this counter chain is also used to time the real time interrupt and the COP circuit.

In addition, counter 2 can be cleared by software. This is accomplished by writing any value to the counter register. This forces a hardware reset of the counter, regardless of the value written. This feature is not available on counter 1 because counter 1 is also used to time the real time interrupt and the COP circuit.

6.1.2 Prescalers

There are two prescalers, one for each of the timer counters. All of the prescaler select bits are included in one register (TPRE). This allows the user to set both prescalers simultaneously so that both counters may be kept synchronized. The external clock edge selects for counter 2 are also included in this register.

Each prescaler is a 3-stage divider with the E-clock as its input, and each stage divides by two. The prescaler output is selectable as E, E/2, E/4, or E/8. After reset, the MC68HC11G5 is configured to use the E-clock as the input to both counters. Initialization software may optionally reconfigure counter 1 to use one of the three prescale taps so that the E-clock is divided by 2, 4, or 8 before driving the free running counter. Initialization software may, at the same time, optionally reconfigure counter 2 to use one of the three prescale taps so that the E-clock is divided by 2, 4, or 8 before driving the free-running counter, and may select the falling, rising or both edges of an external clock. If counter 2 uses the external clock, then the prescaler bits have no effect.

6.1.3 Input Capture Functions

The fixed input capture registers (TIC1 – TIC3) are 16-bit read-only registers which are not affected by reset and are used to latch the value of the free-running counter when a defined transition is sensed by the corresponding input capture edge detector. They optionally cause an interrupt in response to a detected edge event on an input pin. Pairs of bits in the TCTL2 register allow the user to specify that input capture will occur on rising edges only, falling edges only, rising and falling edges or to inhibit captures completely. Registers TMSK1 and TFLG1 provide interrupt enable and interrupt flag bits, respectively Input captures IC1 – IC3 are associated with Port A, bits 2 – 0, respectively.

6.1.4 Output Compare Functions

The fixed output compare registers (TOC1 – TOC4) are 16-bit read/write registers which are initialized to $FFFF by reset and can be used for several purposes. Possible applications include controlling an output waveform and indicating when a period of time has elapsed. The output compare registers are unique in that all bits are readable and writable and are not altered by the timer hardware (except during reset). If an output compare function is not utilized, the unused registers may be used simply as storage locations. Output compares OC1 – OC4 are associated with Port A bits 7 – 4, respectively.

The output compare functions referred to above cause an output action and/or an interrupt when there is a match between a 16-bit output compare register and the free-running counter. At any time the bits controlled by the output compare functions may be read to determine the current state of the output pin.

Three of the fixed output compare functions (OC2, OC3 and OC4) and all of the programmable ones allow software to specify that the corresponding output pin should toggle, be set to one, be cleared to zero, or be disconnected from the output pin logic. This is controlled using the OMx and OML bits in the timer control registers TCTL1 and TCTL3.

Unlike the other output compare functions, OC1 can automatically affect any or all of the Port A output pins associated with OC2 – OC5 (with OC5/IC4 configured for output compare), as a result of a successful compare between the OC1 register and the 16-bit free running counter. Two 5-bit registers are used in conjunction with this function, the output compare 1 mask register (OC1M) and the output compare 1 data register (OC1D). The OC1M register specifies bit-for-bit which Port A lines (bits 3 – 7) are to be affected and the OC1D register specifies what data is to be placed on the affected lines. By configuring the system so that both OC1 and another output compare function both control the same output line, output pulses can be generated with durations as short as one free running counter count.

Since the four other output compare functions (OC2 – OC5) optionally affect four bits of Port A and the pulse accumulator optionally uses bit 7 of Port A, there is an overlap of these functions with the OC1 output compare function. Pins are made available to the OC1 function when other timer functions are not used or are used in a manner that does not require use of the I/O pin.

A write-only register (CFORC) is provided to allow the equivalent of a forced output compare function. Writing a one to any bit in this register immediately has the same effect as a successful comparison on the corresponding output compare function.

In some cases the values in the output compare register and the output action control bits must be changed after each successful comparison to control an output waveform or to establish a new elapsed timeout.

Due to the prescaler, the counter may not change value on each cycle of the E-clock and the comparison may be true for several consecutive E-clock cycles. In order to avoid multiple output actions, the output action is permitted to occur only during the E-clock low time immediately following the cycle during which the match first became true.

An interrupt can also accompany a successful output compare provided that the corresponding interrupt enable bit (OCxI) in the TMSK1 register is set. In this event, the corresponding interrupt flag bit (OCxI) in the TFLG1 register will be set.

After writing to the TOCx register's most significant byte, the output compare function is inhibited for one E-clock cycle, to allow writing two consecutive bytes before making the next comparison. If both bytes of the register are to be changed, a double byte write instruction should be used in order to take advantage of the compare inhibit feature.

MPU writes can be made to either byte of the output compare register without affecting the other byte. When a compare occurs, the output action is taken regardless of whether or not the output compare flag (OCxF) was previously set.

6.1.5 Programmable Input Capture/Output Compares

There are three programmable input capture/output compares. Each channel has a 16-bit register (TO5I4, TO6I5, TO7I6) associated with it which acts as either an output compare register or an input capture register depending on which is specified. OC5/IC4 and OC6/IC5 are associated with counter 1 and OC7/IC6 is associated with counter 2. The selection of output compare or input capture for these functions is controlled by the I4/O5 bit in the PACTL register and the I5/O6

and I6/O7 bits in the TCTL4 register. Output compare actions are controlled by bits in TCTL1, TCTL3 and CFORC. Input capture control is via registers TCTL2 and TCTL3. Registers TMSK1 and TMSK2 provide interrupt enable bits for input capture and output compare functions. TFLG1 and TFLG2 provide corresponding interrupt flag bits.

6.2 REAL TIME INTERRUPT

The real time interrupt feature on the MC68HC11G5 is configured and controlled using three bits (RTR2, RTR1 and RTR0) in the PACTL register to select one of eight interrupt rates. The RTII bit in the TMSK2 register enables the interrupt capability. 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 real time interrupt period elapses before the RTIF flag is set for the first time.

6.3 PULSE ACCUMULATOR

The pulse accumulator is an 8-bit read/write counter (PACNT) which can operate in either of two modes (external event counting or gated time accumulation) depending on the state of the PAMOD control bit in the PACTL register. In the event counting mode, the 8-bit counter is clocked to increasing values by an external pin input. The maximum clocking rate for the external event counting mode is E divided by 2. In the gated time accumulation mode a free-running E/64 clock drives the 8-bit counter, but only while the external PAI input pin is in a selected state.

Although Port A bit 7 would normally be configured as an input when being used for the pulse accumulator, it still drives the pulse accumulator system even when it is configured for use in its other functions. DDRA bit 7 controls the data direction of Port A bit 7. When DDRA bit 7 is zero, Port A bit 7 is an input only pin, unless OC1 is configured to control the pin. When DDRA bit 7 is a one, Port A bit 7 is an output but is still connected to the pulse accumulator input.

PAEN in the PACTL register enables/disables the pulse accumulator.

PEDGE in the PACTL register selects the active edge of the input signal on the PAI pin.

The PAOVF status bit in the TFLG2 register is set when the counter overflows from $FF to $00.

The PAOVI mask bit in the TMSK2 register controls whether or not a hardware interrupt will be generated when the PAOVF flag bit is set.

The PAIF flag bit in the TFLG2 register is set when an active edge is detected on the PAI input pin.

The PAII mask bit in the TMSK2 register controls whether or not a hardware interrupt will be generated when the PAIF status bit is set.

6.4 TIMER, RTI AND PULSE ACCUMULATOR REGISTERS

6.4.1 Count Registers (TCNT1 and TCNT2)

	7	6	5	4	3	2	1	0	
$1**0**0E	BIT15	BIT14	BIT13	BIT12	BIT11	BIT10	BIT9	BIT8	TCNT1
$1**0**0F	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0	
RESET:	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	

READ: Any time.

WRITE: Has no effect for SMOD = 0; forces to $FFF8 for SMOD = 1.

RESET: $0000.

	7	6	5	4	3	2	1	0	
$1**0**52	BIT15	BIT14	BIT13	BIT12	BIT11	BIT10	BIT9	BIT8	TCNT2
$1**0**53	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0	
RESET:	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	

READ: Any time.

WRITE: Forces to $0000 for SMOD = 0; forces to $FFF8 for SMOD = 1.

RESET: $0000.

TCNT1 is associated with OC1 – OC4, OC5/IC4 and OC6/IC5. Note thatdue to the periodic interrupt and COP timer also using this counter as their clock source, an external clock cannot be used to drive this timer counter.

TCNT2 is associated with IC1 – IC3 and OC7/IC6. This counter differs from counter 1 in two ways: it can be driven by an external clock; and it can be reset under software control.

A full counter read should first address the most significant byte. Reading this address causes the least significant byte to be latched into a buffer for the next CPU cycle so that a double byte read will return the full 16-bit state of the counter at the time of the most significant byte read cycle. This buffer is not affected by reset and is accessed when reading the least significant byte of the counter. For double byte read instructions, these two accesses occur on consecutive bus cycles. The buffer is transparent except for the cycle following a most significant byte read so that reads of the least significant byte alone will return the current state of the counter. Software can read the counter at any time without affecting its value. Note that the counter is clocked and read during opposite half cycles of the E-clock.

Both counters are cleared to $0000 during reset. In the normal operating modes (SMOD = 0), writing to TCNT2 causes TCNT2 to be reset to $0000, whereas writing to TCNT1 has no effect. In the test and bootstrap modes only (SMOD = 1), any write to either counter's most significant byte causes

the counter to be preset to $FFF8 regardless of the data written. This preset capability is intended only for factory testing.

Because its width is 16 bits, the value in the free running counter repeats every 65,536 counts (prescaler timeouts). When the count changes from $FFFF to $0000 the timer overflow flag bit (TOxF), in the TFLG2 register, is set. A timer overflow interrupt can be enabled by setting its interrupt enable bit (TOxI), in the TMSK2 register.

6.4.2 Prescaler Register (TPRE)

	7	6	5	4	3	2	1	0	
$1058	TEDGB	TEDGA	PR2B	PR2A	0	0	PR1B	PR1A	TPRE
RESET:	0	0	0	0	0	0	0	0	

READ: Any time.

WRITE: If SMOD = 0, may be written only once during the first 64 cycles. After this time, the bits become read-only. If SMOD = 1, writing is always permitted.

TEDGB, TEDGA — Timer External Clock Edge Select

These two bits determine which edge(s) of the external clock will cause timer counter 2 to increment. The external clock comes in via Port J, bit 0.

TEDGB	TEDGA	Configuration
0	0	Counter 2 uses the internal clock and prescaler
0	1	Count on rising edges only
1	0	Count on falling edges only
1	1	Count on any edge (rising or falling)

Note: the maximum frequency of the input clock must be less than E/2 when counting on only rising or falling edges and must be less than E/4 when counting on both edges.

PR2B, PR2A — Timer Prescaler Select

These bits specify the number of divide-by-2 stages to be inserted between the E-clock and timer counter 2. Note: these bits have no effect if the external clock is selected (TEDGB or TEDGA = 1).

PR2B	PR2A	Prescale Factor
0	0	1
0	1	2
1	0	4
1	1	8

PR1B, PR1A — Timer Prescaler select

These bits specify the number of divide-by-2 stages to be inserted between the E-clock and timer counter 1.

PR1B	PR1A	Prescale Factor
0	0	1
0	1	2
1	0	4
1	1	8

6.4.3 Input Capture Registers (TIC1 – TIC3)

	7	6	5	4	3	2	1	0	
$1010	BIT15	BIT14	BIT13	BIT12	BIT11	BIT10	BIT9	BIT8	TIC1
$1011	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0	
$1012	BIT15	BIT14	BIT13	BIT12	BIT11	BIT10	BIT9	BIT8	TIC2
$1013	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0	
$1014	BIT15	BIT14	BIT13	BIT12	BIT11	BIT10	BIT9	BIT8	TIC3
$1015	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0	

READ: Any time.

WRITE: Has no meaning or effect.

RESET: Indeterminate (not initialized).

These registers are used to latch the value of timer counter 2 when a defined transition is sensed by the corresponding input capture edge detector. The result obtained by an input capture corresponds to the value of the counter one cycle after the transition which triggered the edge detection logic. This delay is required for internal synchronization.

After a read of the most significant byte of the register, counter transfer is inhibited during the next E-clock cycle. During double byte reads, this inhibited transfer will occur between consecutive read cycles of the most and least significant bytes.

6.4.4 Output Compare Registers (TOC1 – TOC4)

	7	6	5	4	3	2	1	0	
$1016	BIT15	BIT14	BIT13	BIT12	BIT11	BIT10	BIT9	BIT8	TOC1
$1017	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0	
$1018	BIT15	BIT14	BIT13	BIT12	BIT11	BIT10	BIT9	BIT8	TOC2
$1019	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0	
$101A	BIT15	BIT14	BIT13	BIT12	BIT11	BIT10	BIT9	BIT8	TOC3
$101B	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0	
$101C	BIT15	BIT14	BIT13	BIT12	BIT11	BIT10	BIT9	BIT8	TOC4
$101D	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0	

READ: Any time.

WRITE: Any time.

RESET: $FFFF

A write to a high order byte inhibits the compare for the next E-clock cycle. When the TOCx register matches the TCNT1 register, the OCxF bit in the TFLG1 register is set and the specified output action takes place.

All output compare registers have a separate dedicated comparator for comparing against the free running counter. If a match is found, the corresponding output compare flag bit (OCxF) is set and a specified action is automatically taken.

For output compare functions OC2, OC3 and OC4, the automatic action is controlled by pairs of bits in the TCTL1 control register. Each pair of control bits is encoded to specify the output action to be taken as a result of a successful OCx compare. Output compare functions OC2, OC3 and OC4 are always associated with Port A bits 6 – 4 respectively.

6.4.5 Output Compare 5/Input Capture 4 Register (TO5I4)

This is a shared register which acts as the output compare OC5 register or as the input capture IC4 register depending on the state of the I4/O5 bit in the PACTL register. This register is associated with timer counter 1 and with Port A, bit 3.

	7	6	5	4	3	2	1	0	
$1**0**1E	BIT15	BIT14	BIT13	BIT12	BIT11	BIT10	BIT9	BIT8	TO5I4
$1**0**1F	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0	
RESET:	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	

READ: Any time.

WRITE: Any time this register is configured for OC operation.

RESET: $FFFF

6.4.6 Output Compare 6/Input Capture 5 Register (TO6I5)

This is a shared register which acts as the output compare OC6 register or as the input capture IC5 register depending on the state of the I5/O6 bit in the TCTL4 register. This register is associated with timer counter 1 and with Port J, bit 1.

	7	6	5	4	3	2	1	0	
$1**0**54	BIT15	BIT14	BIT13	BIT12	BIT11	BIT10	BIT9	BIT8	TO6I5
$1**0**55	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0	
RESET:	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	

READ: Any time.

WRITE: Any time this register is configured for OC operation.

RESET: $FFFF

6.4.7 Output Compare 7/Input Capture 6 Register (TO7I6)

This is a shared register which acts as the output compare OC7 register or as the input capture IC6 register depending on the state of the I6/O7 bit in the TCTL4 register. This register is associated with timer counter 2 and with Port J, bit 2.

	7	6	5	4	3	2	1	0	
$1056	BIT15	BIT14	BIT13	BIT12	BIT11	BIT10	BIT9	BIT8	TO7I6
$1057	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0	
RESET:	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	

READ: Any time.

WRITE: Any time this register is configured for OC operation.

RESET: $FFFF

There are both input control bits and output control bits associated with these functions. Only the bits associated with the selected function (input or output) will affect the operation of the timer channel.

6.4.8 Output Compare 1 Action Mask Register (OC1M)

	7	6	5	4	3	2	1	0	
$100C	OC1M7	OC1M6	OC1M5	OC1M4	OC1M3	0	0	0	OC1M
RESET:	0	0	0	0	0	0	0	0	

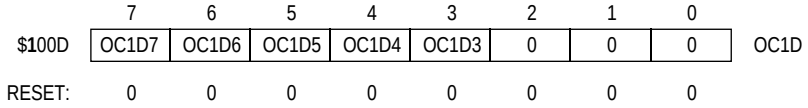
READ: Any time (bits 2 through 0 always return 0).

WRITE: Any time (writes to bits 2 through 0 have no meaning or effect).

RESET: $00 (OC1 disconnected from Port A logic).

OC1M is used to specify which bits of Port A (I/O and timer port) are to be affected as a result of a successful OC1 compare. The bits of OC1M correspond bit-for-bit with the bits of Port A. For each bit to be affected by the successful OC1 compare, the corresponding bit in OC1M must be set to one. Each Port A line associated with an OC1Mx bit which is set to one will be forced to be an output regardless of the state of the associated DDRA bit. This does not change the state of the DDRA bit.

6.4.9 Output Compare 1 Action Data Register (OC1D)



	7	6	5	4	3	2	1	0	
$100D	OC1D7	OC1D6	OC1D5	OC1D4	OC1D3	0	0	0	OC1D
RESET:	0	0	0	0	0	0	0	0	

READ: Any time (bits 2 – 0 always return 0).

WRITE: Any time (writes to bits 2 – 0 have no meaning or effect).

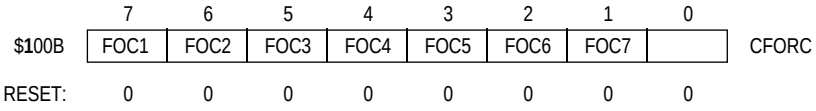
RESET: $00

OC1D is used to specify the data to be written to the affected bits of Port A as the result of a successful OC1 compare. The bits of OC1D correspond bit-for-bit with the bits of Port A (bits 3 – 7). When a successful OC1 compare occurs, for each bit that is set in OC1M, the corresponding data bit in OC1D is written to the corresponding bit of Port A. If there is a conflicting situation where an OC1 compare and another output compare function occur during the same E-clock cycle, and both attempt to alter the same Port A bit, the OC1 action will take priority.

One reason for providing this special capability on OC1 is to allow control of multiple I/O pins automatically with a single output compare function. For example, if the OC2 and OC3 functions are being used for internal timing functions, their associated Port A pins are free to be used for other purposes. These two pins could be controlled simultaneously as high speed timed outputs using the OC1 function by setting the two corresponding bits in the OC1M register to one.

The special I/O pin control on the OC1 function also allows more than one output compare function to control a single I/O pin. For example, the OC1 function could be configured to affect only bit 3 of Port A (by setting OC1M = $08). The OC5 function could set Port A bit 3 to a logic one and the OC1 function could reset it to a logic low on the very next count of the free-running counter.

6.4.10 Compare Force Register (CFORC)



	7	6	5	4	3	2	1	0	
$100B	FOC1	FOC2	FOC3	FOC4	FOC5	FOC6	FOC7		CFORC
RESET:	0	0	0	0	0	0	0	0	

READ: Any time but will always return $00 (1 state is transient).

WRITE: Any time (writes to bit 0 have no meaning or effect).

RESET: $00 (no actions forced).

FOC1 to FOC7 — Force Output Compare “x” Action

Writing a one to bit “x” in this register causes the action which is programmed for output compare “x” to occur at the next transition of the prescaled timer clock. The action taken is the same as if a successful comparison had just taken place with the TOCx register, with the exception that the interrupt flag is not set.

6.4.11 Control Register 1 (TCTL1)

	7	6	5	4	3	2	1	0	
$1020	OM2	OL2	OM3	OL3	OM4	OL4	OM5	OL5	TCTL1
RESET:	0	0	0	0	0	0	0	0	

READ: Any time.

WRITE: Any time.

RESET: $00

OMx — Output Mode; OLx — Output Level

These four pairs of control bits are encoded to specify the output action to be taken as a result of a successful OCx compare (OC2 – OC5). When either OMx or OLx is set, the pin associated with OCx becomes an output tied to OCx regardless of the state of the associated DDR bit. Output compare OC5 only functions if the TO5I4 register is programmed for output compare OC5 operation via the I4/O5 bit in the PACTL register.

OMx	OLx	Action taken upon successful compare
0	0	Timer disconnected from output pin logic
0	1	Toggle OCx output line
1	0	Clear OCx output line to zero
1	1	Set OCx output line to one

6.4.12 Timer Control Register 2 (TCTL2)

	7	6	5	4	3	2	1	0	
$1021	EDG4B	EDG4A	EDG1B	EDG1A	EDG2B	EDG2A	EDG3B	EDG3A	TCTL2
RESET:	0	0	0	0	0	0	0	0	

READ: Any time

WRITE: Any time

RESET: $00

EDGxB, EDGxA — Input Capture Edge Control

The level transition which triggers counter transfer is defined by the corresponding input edge bits (EDGxB, EDGxA). These bit pairs are encoded to configure input captures IC1 – IC4 to occur on rising edges, falling edges, either edge, or to inhibit capture. Input capture 4 only functions if the TO5I4 register is programmed for input capture IC4 operation by the I4/O5 bit in the PACTL register.

EDGxB	EDGxA	Configuration
0	0	Capture disabled
0	1	Capture on rising edges only
1	0	Capture on falling edges only
1	1	Capture on any edge (rising or falling)

Note: input captures do not force the direction of the associated port. If the port bit associated with an input capture is programmed to be an output, then input captures will occur on the appropriate changes of state caused by writing to the port.

6.4.13 Timer Control Register 3 (TCTL3)

	7	6	5	4	3	2	1	0	
$1050	EDG5B	EDG5A	EDG6B	EDG6A	OM6	OL6	OM7	OL7	TCTL3
RESET:	0	0	0	0	0	0	0	0	

READ: Any time.

WRITE: Any time.

RESET: $00

EDGxB, EDGxA — Input Capture Edge Control

These control bits configure the input capture edge detector circuits for input captures IC5 and IC6.

EDGxB	EDGxA	Configuration
0	0	Capture disabled
0	1	Capture on rising edges only
1	0	Capture on falling edges only
1	1	Capture on any edge (rising or falling)

Input capture IC5 only functions if the TO6I5 register is programmed for input capture IC5 operation by the I5/O6 bit in the TCTL4 register being set to one. Input capture IC6 functions only if the TO7I6 register is programmed for input capture IC5 operation by the I6/O7 bit in the TCTL4 register being set to one.

OMx, OLx — Output Mode, Output Level

These control bits are encoded to specify the output action to be taken as a result of successful compares on OC6 and OC7. When either OMx or OLx is set to one, the pin associated with OCx becomes an output tied to OCx, regardless of the state of the associated DDR bit.

OMx	OLx	Action taken upon successful compare
0	0	Timer disconnected from output pin logic
0	1	Toggle OCx output line
1	0	Clear OCx output line to zero
1	1	Set OCx output line to one

Output compare OC6 only functions if the TO6I5 register is programmed for output compare OC6 operation by the I5/O6 bit in the TCTL4 register being set to zero. Output compare OC7 functions only if the TO7I6 register is programmed for output compare OC7 operation by the I6/O7 bit in the TCTL4 register being set to zero.

6.4.14 Control Register 4 (TCTL4)

	7	6	5	4	3	2	1	0	
$1051	CT2SP	CT1SP	0	0	0	0	I5/O6	I6/O7	TCTL4
RESET:	0	0	0	0	0	0	0	0	

READ: Any time (bits 2 – 7 always read 0 if not in test mode).

WRITE: Any time (writes to bits 2 – 7 have no meaning or effect if not in test or bootstrap mode).

RESET: $00

CT2SP — Timer Counter 2 Stop (used to facilitate testing)

0 – Timer counter 2 is free-running.

1 – Timer counter 2 is stopped if in test or bootstrap mode (SMOD = 1).

CT1SP — Timer Counter 1 Stop (used to facilitate testing)

0 – Timer counter 1 is free-running.

1 – Timer counter 1 is stopped if in test or bootstrap mode (SMOD = 1).

I5/O6 — Configure TO6I5 Register for Input Capture or Output Compare

0 – Output compare 6 function enabled (no IC5).

1 – Input capture 5 function enabled (no OC6).

I6/O7 — Configure TO7I6 Register for Input Capture or Output Compare

0 – Output compare 7 function enabled (no IC6).

1 – Input capture 6 function enabled (no OC7).

6.4.15 Main Timer Interrupt Mask Register 1 (TMSK1)

The bits in TMSK1 correspond bit-for-bit with the bits in the TFLG1 status register. A zero disables the corresponding flag from causing a hardware interrupt. A one enables the corresponding flag to cause a hardware interrupt.

	7	6	5	4	3	2	1	0	
$1022	OC1I	OC2I	OC3I	OC4I	4/5I	IC1I	IC2I	IC3I	TMSK1
RESET:	0	0	0	0	0	0	0	0	

READ: Any time

WRITE: Any time

RESET $00

OC1I, OC2I, OC3I, OC4I — Output Compare “x” Interrupt Enable

0 – Interrupt inhibited.

1 – OCx interrupt requested when OCxF flag set

4/5I — Input Capture 4/Output Compare 5 Interrupt Enable

0 – Interrupt inhibited.

1 – IC4 or OC5 interrupt requested when 4/5F flag set

IC1I, IC2I, IC3I — Input Capture “x” Interrupt Enable

0 – Interrupt inhibited.

1 – ICx interrupt requested when ICxF flag set

6.4.16 Main Timer Interrupt Flag Register 1 (TFLG1)

The main timer system flag register (TFLG1) contains flag bits which are set by hardware when the corresponding timer interrupt condition occurs. Any flag bits in the TFLG1 register which are set will remain set until they are cleared by writing ones to those bits.

	7	6	5	4	3	2	1	0	
$1023	OC1F	OC2F	OC3F	OC4F	4/5F	IC1F	IC2F	IC3F	TFLG1
RESET:	0	0	0	0	0	0	0	0	

READ: Any time.

WRITE: Used in clearing mechanism (writing ones to bits set cause these bits to be cleared).

RESET: $00

OC1F, OC2F,OC3F, OC4F — Output Compare “x” Flag

Set when the 16-bit timer counter register matches the OCx compare register. These bits are cleared by writing to the TFLG1 register with the corresponding bits (4–7) set.

4/5F — Input Capture 4/Output Compare 5 Flag

Set when an input capture occurs on IC4 or an output compare occurs on OC5. This bit is cleared by writing to the TFLG1 register with bit 3 set.

IC1F, IC2F, IC3F — Input Capture “x” Flag

Set when an input capture occurs on ICx. These bits are cleared by writing to the TFLG1 register with the corresponding bits (0–2) set.

6.4.17 Miscellaneous Timer Interrupt Mask Register 2 (TMSK2)

The bits in TMSK2 correspond bit-for-bit with the bits in the TFLG2 status register. A zero inhibits the corresponding flag from causing a hardware interrupt. A one enables the corresponding flag to cause a hardware interrupt.

	7	6	5	4	3	2	1	0	
$1024	TO1I	RTII	PAOVI	PAII	TO2I	5/6I	6/7I	0	TMSK2
RESET:	0	0	0	0	0	0	0	0	

READ: Any time.

WRITE: Any time.

RESET: $00

TO1I — Timer Overflow 1 Interrupt Enable

0 – Interrupt inhibited.

1 – Hardware interrupt requested when TO1F flag set.

RTII — RTI Interrupt Enable

0 – Interrupt inhibited.

1 – Hardware interrupt requested when RTIF flag set.

PAOVI — Pulse Accumulator Overflow Interrupt Enable

0 – Interrupt inhibited

1 – Hardware interrupt requested when PAOVF flag set

PAII — Pulse Accumulator Input Interrupt Enable

0 – Interrupt inhibited

1 – Hardware interrupt requested when PAIF flag set

TO2I — Timer Overflow 2 Interrupt Enable

0 – Interrupt inhibited

1 – Hardware interrupt requested when TO2F flag set

5/6I — Input Capture 5/Output Compare 6 Interrupt Enable

0 – Interrupt inhibited

1 – Hardware interrupt requested when 5/6F flag set

6/7I — Input Capture 6/Output Compare 7 Interrupt Enable

0 – Interrupt inhibited

1 – Hardware interrupt requested when 6/7F flag set

6.4.18 Miscellaneous Timer Interrupt Flag Register 2 (TFLG2)

The miscellaneous timer system flag register (TFLG2) contains flag bits which are set by hardware when the corresponding timer interrupt condition occurs. Any flag bits in the TFLG1 register which are set will remain set until they are cleared by writing ones to those bits.

	7	6	5	4	3	2	1	0	
$1025	TO1F	RTIF	PAOVF	PAIF	TO2F	5/6F	6/7F	0	TFLG2
RESET:	0	0	0	0	0	0	0	0	

READ: Any time.

WRITE: Used in clearing mechanism (see above).

RESET: $00

TO1F — Timer Overflow 1 Flag

Set when 16-bit free running timer 1 overflows from $FFFF to $0000. This bit is cleared by writing to the TFLG2 register with bit 7 set.

RTIF — Real Time (Periodic) Interrupt Flag

Set when the tap point selected becomes set. This bit is cleared by writing to the TFLG2 register with bit 6 set.

PAOVF — Pulse Accumulator Overflow Flag

Set when the 8-bit pulse accumulator overflows from $FF to $00. This bit is cleared by writing to the TFLG2 register with bit 5 set.

PAIF — Pulse Accumulator Input Edge Flag

Set when the selected edge is detected at the PAI input pin. In event mode the event edge triggers PAIF and in gated time accumulation mode the trailing edge of the gate signal at the PAI input pin triggers PAIF. This bit is cleared by writing to the TFLG2 register with bit 4 set.

TO2F — Timer Overflow 2 Flag

Set when 16-bit free-running timer 2 overflows from $FFFF to $0000. This bit is cleared by writing to the TFLG2 register with bit 3 set.

5/6F — Input Capture 5/Output Compare 6 Flag

Set by input capture IC5 or output compare OC6, depending on which function is selected. This bit is cleared by writing to the TFLG2 register with bit 2 set.

6/7F — Input Capture 6/Output Compare 7 Flag

Set by input capture IC6 or output compare OC7, depending on which function is selected. This bit is cleared by writing to the TFLG2 register with bit 1 set.

6.4.19 Count Register (PACNT)

PACNT is a read/write 8-bit counter register which is not initialized by reset. The PACTL register contains four control bits which enable and configure the pulse accumulator system. The pulse accumulator uses Port A bit 7 as its PAI input but this pin can also serve as a general purpose I/O pin and as the timer output compare OC1 output.

	7	6	5	4	3	2	1	0	
$1027	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0	PACNT
RESET:	U	U	U	U	U	U	U	U	

READ: Any time (returns count from pulse accumulator counter).

WRITE: Any time.

RESET: Indeterminate.

6.4.20 Pulse Accumulator Control Register (PACTL)

	7	6	5	4	3	2	1	0	
$1026	0	PAEN	PAMOD	PEDGE	I4/O5	RTR2	RTR1	RTR0	PACTL
RESET:	0	0	0	0	0	0	0	0	

READ: Any time.

WRITE: Any time.

RESET: $00

PAEN — Pulse Accumulator System Enable

When PAEN is zero, the pulse accumulator counter is disabled from counting and the PAIF and PAOVF flags cannot be set. The counter value and the states of the two flags are not altered by the state of the PAEN bit (writing PAEN to zero does not clear them). When PAEN is a one, the pulse accumulator counter and the setting mechanisms on the two pulse accumulator flags PAIF and PAOVF are enabled.

0 – Pulse accumulator system disabled (no flags can become set and the counter is stopped).

1 – Pulse accumulator system enabled

PAMOD — Pulse Accumulator Mode

The PAMOD control bit specifies “event” or “gated time accumulation” mode.

0 – Event counter mode.

1 – Gated time accumulation mode.

PEDGE — Pulse Accumulator Edge Control

The PEDGE bit is used to specify the active edge for the PAI input pin which is interpreted as the trailing edge for the PAI gate enable input when the system is configured for gated time accumulation.

PEDGE in Event Counter Mode (PAMOD = 0):

0 – Falling edges on PAI pin cause the count to be incremented.

1 – Rising edges on PAI pin cause the count to be incremented.

PEDGE in Gated Time Accumulation Mode (PAMOD = 1):

0 – PAI input pin high enables E divided by 64 clock to pulse accumulator and the trailing falling edge on PAI sets the PAIF flag.

1 – PAI input pin low enables E divided by 64 clock to pulse accumulator and the trailing rising edge on PAI sets the PAIF flag.

I4/O5 — Configure TO5I4 register for input capture or output compare

0 – Output compare 5 function enabled (no IC4).

1 – Input capture 4 function enabled (no OC5).

RTR2, RTR1, RTR0 — RTI Interrupt Rate

These bits select one of six rates for the real time periodic interrupt circuit (see Table 6-1). Reset clears these bits and after reset a full RTI period elapses before the first RTI interrupt occurs.

Table 6-1. Real Time Interrupt Rates

RTR2	**RTR1**	**RTR0**	**Divide E By**	**XTAL = 2^{23}**	**XTAL = 8.0MHz**	**XTAL = 4.9152MHz**	**XTAL = 4.0MHz**	**XTAL = 3.6864MHz**
1	1	0	2^{11}	0.98 ms	1.02 ms	1.67 ms	2.05 ms	2.22 ms
1	1	1	2^{12}	1.95 ms	2.05 ms	3.33 ms	4.10 ms	4.44 ms
0	0	0	2^{13}	3.91 ms	4.10 ms	6.67 ms	8.19 ms	8.89 ms
0	0	1	2^{14}	7.81 ms	8.19 ms	13.33 ms	16.38 ms	17.78 ms
0	1	0	2^{15}	15.62 ms	16.38 ms	26.67 ms	32.77 ms	35.56 ms
0	1	1	2^{16}	31.25 ms	32.77 ms	53.33 ms	65.54 ms	71.11 ms
			E =	2.1MHz	2.0MHz	1.2288MHz	1.0MHz	921.6kHz

SECTION 7

SERIAL COMMUNICATIONS INTERFACE

This section describes the UART type serial communications interface system (SCI). The SCI can be used, for example, to connect a CRT terminal or personal computer to the MCU or to form a serial communication network connecting several widely distributed MCUs. It should be noted that the SCI is one of two independent serial I/O subsystems in the MC68HC11G5. The other serial I/O system on the MC68HC11G5, the serial peripheral interface (SPI), provides for high speed synchronous serial communication to peripherals or other MCUs (usually located on the same PC board as the MC68HC11G5).

7.1 OVERVIEW AND FEATURES

The SCI on the MC68HC11G5 is a full duplex UART type asynchronous system. The SCI uses standard non-return-to-zero (NRZ) format (one start bit, eight or nine data bits, and one stop bit). An on-chip baud rate generator derives standard baud rate frequencies from the MCU oscillator. Both the transmitter and the receiver are double buffered so back-to-back characters can be handled easily, even if the CPU is delayed in responding to the completion of an individual character. The SCI transmitter and receiver are functionally independent but use the same data format and baud rate.

7.1.1 SCI Two-wire System Features:

- Standard NRZ (mark/space) format.
- Advanced error detection method includes noise detection for noise duration of up to 1/16th bit time.
- Full-duplex operation.
- Software programmable for one of 32 different baud rates.
- Software selectable word length (eight or nine bit words).
- Separate transmitter and receiver enable bits.
- Capable of being interrupt driven.
- Four separate enable bits available for interrupt control.

7.1.2 SCI Receiver Features

- Receiver wake-up function (idle line or address bit).
- Idle line detect.

- Framing error detect.
- Noise detect.
- Overrun detect.
- Receiver data register full flag.

7.1.3 SCI Transmitter Features

- Transmit data register empty flag.
- Transmit complete flag.
- Send break.

7.2 FUNCTIONAL DESCRIPTION

A block diagram of the SCI is shown in Figure 7-1. The user has option bits in serial control register 1 (SCCR1) to select the “wake-up” method (WAKE bit) and data word length (M bit) of the SCI. Serial communications control register 2 (SCCR2) provides control bits which individually enable/disable the transmitter or receiver (TE and RE, respectively), enable system interrupts (TIE, TCIE, ILIE) and provide the wake-up enable bit (RWU) and the send break code bit (SBK). Control bits in the baud rate register (BAUD) allow the user to select one of 32 different baud rates for the transmitter and receiver.

Data transmission is initiated by a write to the serial communications data register (SCDR). Provided the transmitter is enabled, data stored in the SCDR is transferred to the transmit data shift register. This transfer of data sets the transmit data register empty flag (TDRE) in the SCI status register (SCSR) and may generate an interrupt if the transmit interrupt is enabled. The transfer of data to the transmit data shift register is synchronized with the bit rate clock (Figure 7-2). All data is transmitted least significant bit first. Upon completion of data transmission, the transmission complete flag (TC) in the SCSR is set (provided no pending data, preamble or break is to be sent), and an interrupt may be generated if the transmit complete interrupt is enabled. If the transmitter is disabled, and the data, preamble or break (in the transmit data shift register) has been sent, the TC bit will also be set. This will also generate an interrupt if the transmission complete interrupt enable bit (TCIE) is set. If the transmitter is disabled in the middle of a transmission, that character will be completed before the transmitter gives up control of the TXD pin.

When SCDR is read, it contains the last data byte received, provided that the receiver is enabled. The receive data register full flag bit (RDRF) in the SCSR is set to indicate that a data byte has been transferred from the input serial shift register to the SCDR, which can cause an interrupt if the receiver interrupt is enabled. The data transfer from the input serial shift register to the SCDR is synchronized by the receiver bit rate clock. The OR (overrun), NF (noise), or FE (framing) error flags in the SCSR may be set if data reception errors occurred.

An idle line interrupt is generated if the idle line interrupt is enabled and the IDLE bit (which detects idle line transmission) in SCSR is set. This allows a receiver that is not in the wake-up mode to detect the end of a message or the preamble of a new message, or to resychronize with the transmitter. A valid character must be received before the idle line condition or the IDLE bit will not be set and and idle line interrupt will not be generated.

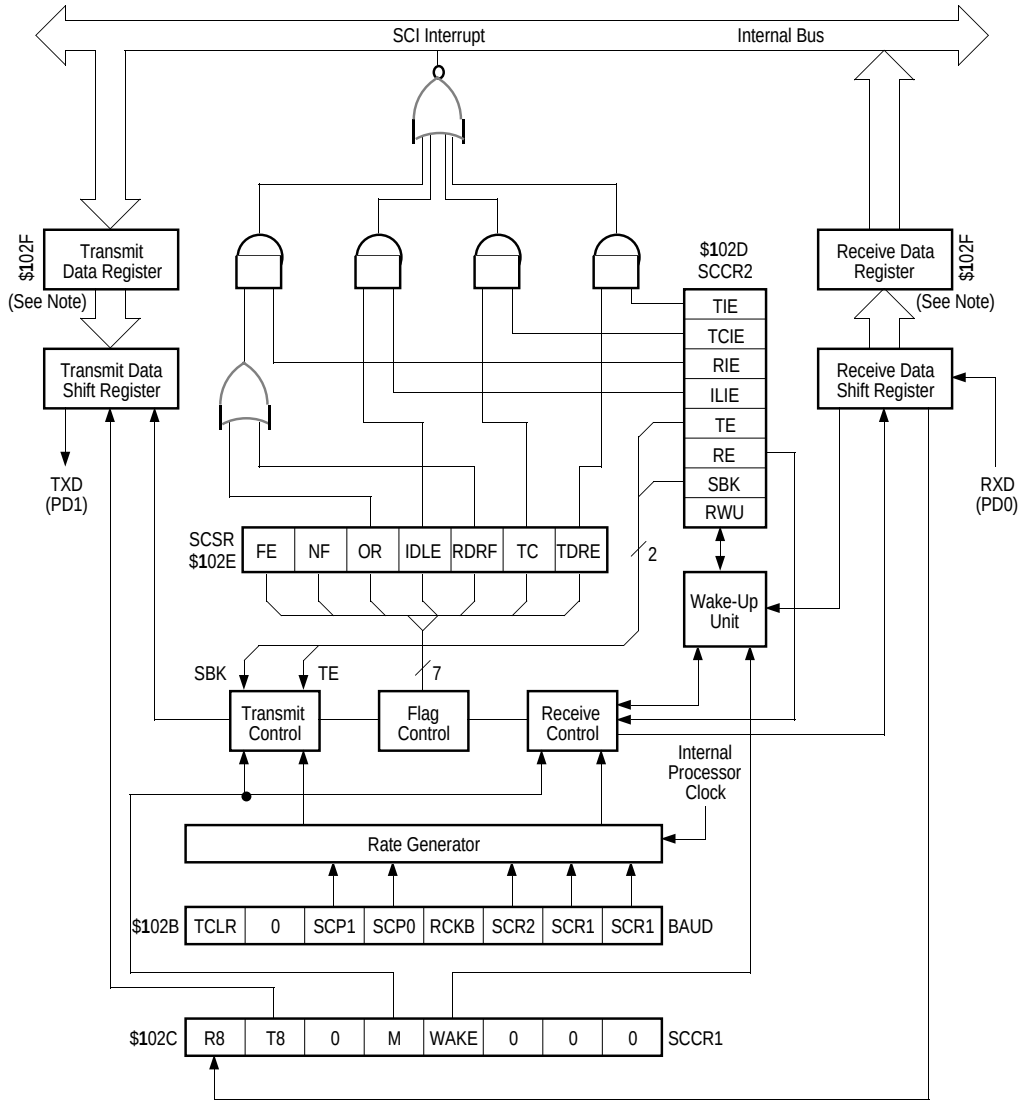


Figure 7-1. Serial Communications Interface Block Diagram

Note: The Serial Communications Data Register (SCDAT) is controlled by the internal R/$\overline{W}$ signal. It is the transmit data register when written and the receive data register when read.

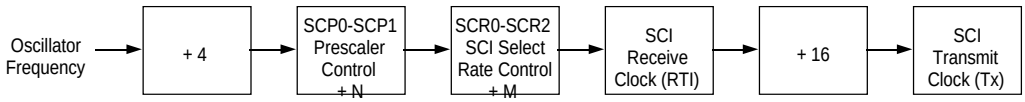


Figure 7-2. Rate Generator Division

7.3 DATA FORMAT

Receive data or transmit data is the serial data that is transferred to the internal data bus from the receive data input pin (RXD) or from the internal bus to the transmit data output pin (TXD). The non-return-to-zero (NRZ) data format shown in Figure 7-3 is used and must meet the following criteria:

1) The idle line is brought to a logic one state prior to transmission/reception of a character.
2) A start bit (logic zero) is used to indicate the start of a frame.
3) The data is transmitted and received least significant bit first.
4) A stop bit (logic one) is 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.
5) A break is defined as the transmission or reception of a low (logic zero) for at least one complete frame time.

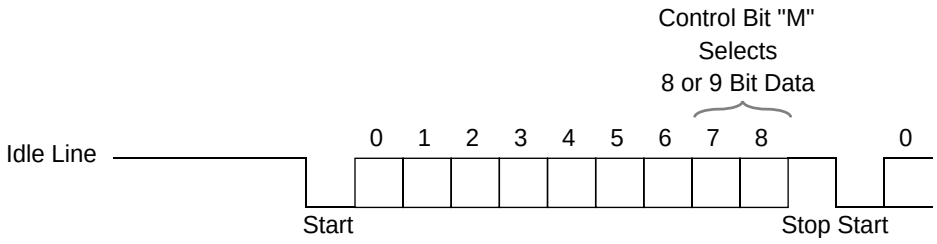


Figure 7-3. Data Format

7.4 RECEIVER WAKE-UP OPERATION

The MC68HC11G5 receiver logic hardware also supports a receiver wake-up function which is intended for systems having more than one receiver. With this function a transmitting device directs messages to an individual receiver or group of receivers by passing addressing information as the initial byte(s) of each message. The wake-up function allows receivers not addressed to remain in in a dormant state for the remainder of the unwanted message. This eliminates any further software overhead to service the remaining characters of the unwanted message and thus improves system performance.

The receiver is placed in wake-up mode by setting the receiver wake-up bit (RWU) in the SCCR2 register. While RWU is set, all of the receiver related status flags (RDRF, IDLE, OR, NF, and FE) are inhibited (cannot become set). Note that the idle line detect function is inhibited while the RWU bit is set. Although RWU may be cleared by a software write to SCCR2, it would be unusual to do so. Normally RWU is set by software and gets cleared automatically with hardware by one of the two methods described below.

7.4.1 Idle Line Wake-up

In idle line wake-up mode, a dormant receiver wakes up as soon as the RXD line becomes idle. Idle is defined as a continuous logic high level on the RXD line for ten (or eleven) full bit times. Systems using this type of wake-up must provide at least one character time of idle between messages to wake up sleeping receivers, but must not allow any idle time between characters within a message.

7.4.2 Address Mark Wake-up

In address mark wake-up, the most significant bit (MSB) in a character is used to indicate that the character is an address (1) or a data (0) character. Sleeping receivers will wake up whenever an address character is received. Systems using this method for wake-up would set the MSB of the first character of each message and leave it clear for all other characters in the message. Idle periods may be present within messages and no idle time is required between messages for this wake-up method.

7.5 RECEIVE DATA (RXD)

Receive data is the serial data that is applied through the input line and the serial communications interface to the internal bus. The receiver circuitry clocks the input at a rate equal to 16 times the baud rate and this time is referred to as the RT clock.

Once a valid start bit is detected the start bit, each data bit and the stop bit are sampled three times at RT intervals 8 RT, 9 RT and 10 RT (1 RT is the position where the bit is expected to start), as shown in Figure 7-4. The value of the bit is determined by voting logic which takes the value of the majority of the samples.

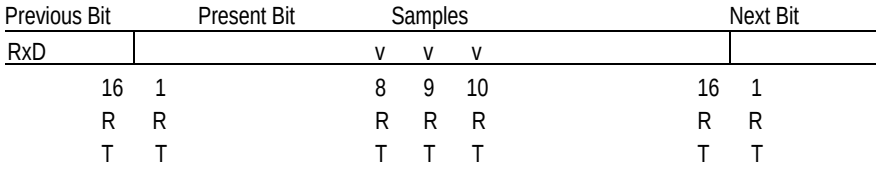


Figure 7-4. Sampling Technique Used On All Bits

7.6 START BIT DETECTION

When the RXD input is detected low, it is tested for three more sample times (referred to as the start edge verification samples in Figure 7-5). If at least two of these three verification samples detect a logic zero, a valid start bit has been detected, otherwise the line is assumed to be idle. A noise flag is set if all three verification samples do not detect a logic zero. A valid start bit could be assumed with a set noise flag present.

If there has been a framing error without detection of a break (10 zeros for 8-bit format or 11 zeros for 9-bit format), the circuit continues to operate as if there actually was a stop bit and the start edge will be placed artificially. The last bit received in the data shift register is inverted to a logic one, and the three logic one start qualifiers (shown in Figure 7-5) are forced into the sample shift register during the interval when detection of a start bit is anticipated (see Figure 7-6); therefore, the start bit will be accepted no sooner than it is anticipated.

If the receiver detects that a break produced the framing error, the start bit will not be artificially induced and the receiver must actually detect a logic one before the start bit can be recognised (see Figure 7-7).

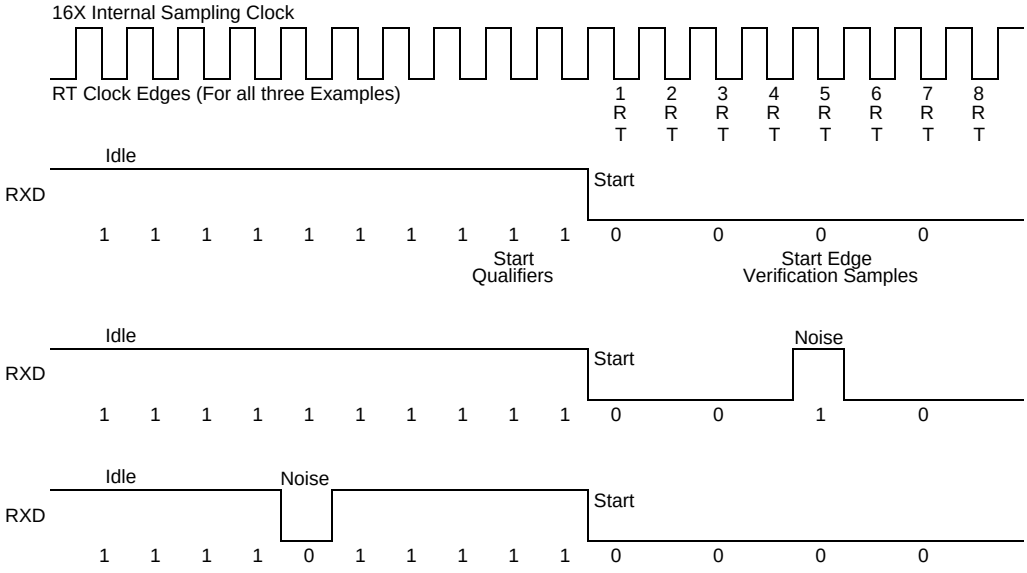
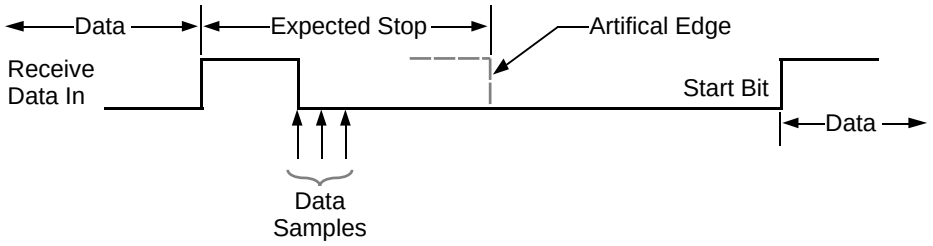
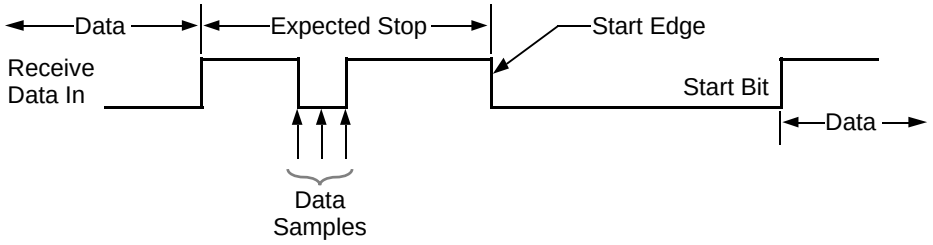


Figure 7-5. Examples of Start Bit Sampling Techniques



(a) Case 1, Receive Line Low During Artificial Edge



(b) Case 2, Receive Line High During Expected Start Edge

Figure 7-6. SCI Artificial Start Following a Framing Error

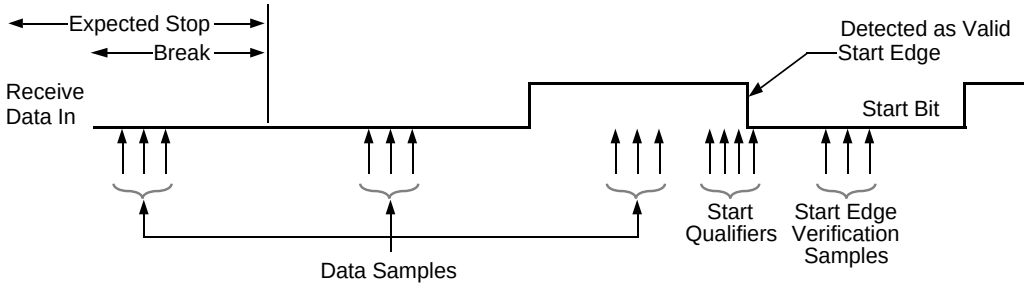


Figure 7-7. SCI Start Bit Following a Break

7.7 TRANSMIT DATA (TXD)

Transmit data is the serial data from the internal data bus that is applied through the serial communications interface to the output line. The transmitter generates a bit time by using a derivative of the RT clock, thus producing a transmission rate equal to 1/16th that of the receiver sample clock.

7.8 SCI REGISTERS

Primarily the SCI system is configured and controlled by five registers BAUD, SCCR1, SCCR2, SCSR, and SCDAT). In addition, the Port D data and data direction registers and the Port D wired-OR mode bit in the SPCR register are secondarily related to the SCI system. Reference should be made to the block diagram shown in Figure 7-1.

7.8.1 Serial Communications Data Register (SCDAT)

The SCI data register (SCDAT) shown in the following figure is actually two separate registers. When SCDAT is read, the read-only receive data register is accessed and when SCDAT is written, the write-only transmit data register is accessed.

	7	6	5	4	3	2	1	0	
$102F	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0	SCDAT
RESET:	0	0	0	0	0	0	0	0	

READ: Accesses the read-only SCI receive data register (RDR).

WRITE: Accesses the write-only SCI transmit data register (TDR).

RESET: Does not affect this address.

7.8.2 Serial Communications Control Register 1 (SCCR1)

The SCI control register 1 (SCCR1) contains control bits related to the nine data bit character format and the receiver wake-up feature. Four of the bits in this register are not used and always read as zeros.

	7	6	5	4	3	2	1	0	
$102C	R8	T8	0	M	WAKE	0	0	0	SCCR1
RESET:	U	U	0	0	0	0	0	0	

R8 — Receive Data Bit 8

READ: Any time.

WRITE: Has no meaning or effect.

This bit is the ninth serial data bit received when the SCI system is configured for nine data bit operation (M = 1). The most significant bit (bit 8) of the received character is transferred into this bit at the same time as the remaining eight bits (bits 0 – 7) are transferred from the serial receive shifter to the SCI receive data register.

T8 — Transmit Data Bit 8

READ: Any time.

WRITE: Any time.

This bit is the ninth data bit to be transmitted when the SCI system is configured for nine data bit operation (M = 1). When the eight low order bits (bits 0 – 7) of a transmit character are transferred from the SCI data register to the serial transmit shift register, this bit (bit 8) is transferred to the ninth bit position of the shifter.

M — Mode (select character format)

READ: Any time.

WRITE: Any time.

0 – 1 start bit, 8 data bits, 1 stop bit.

1 – 1 start bit, 8 data, 9th data bit, 1 stop bit.

The M bit controls the character length for both the transmitter and receiver at the same time. The 9th data bit is most commonly used as an extra stop bit or in conjunction with the “address mark” wake-up method. It can also be used as a parity bit.

WAKE — Wake-up Mode Select

READ: Any time.

WRITE: Any time.

0 – Wake-up on idle line.

1 – Wake-up on address mark.

Note: Bits 2 – 1 are not implemented and always read as zeros.

7.8.3 Serial Communications Control Register 2 (SCCR2)

The SCI control register 2 (SCCR2) provides the control bits that enable/disable individual SCI functions.

	7	6	5	4	3	2	1	0	
$102D	TIE	TCIE	RIE	ILIE	TE	RE	RWU	SBK	SCCR2
RESET:	0	0	0	0	0	0	0	0	

READ: Any time

WRITE: Any time

RESET: $00

TIE — Transmit Interrupt Enable

0 – TDRE interrupts disabled

1 – SCI interrupt if TDRE = 1

TCIE — Transmit Complete Interrupt Enable

0 – TC interrupts disabled

1 – SCI interrupt if TC = 1

RIE — Receiver Interrupt Enable

0 – RDRF and OR interrupts disabled

1 – SCI interrupt if RDRF or OR = 1

ILIE — Idle Line Interrupt Enable

0 – IDLE interrupts disabled.

1 – SCI interrupt if IDLE = 1

TE — Transmitter Enable

When the transmit enable bit is set, the transmit shift register output is applied to the TXD line. Depending on the state of control bit M (SCCR1), a preamble of 10 (M = 0) or 11 (M = 1) consecutive ones is transmitted when software sets the TE bit from a cleared state. After loading the last byte in the serial communications data register and receiving the TDRE flag, the user can clear TE. Transmission of the last byte will then be completed before the transmitter gives up control of the TXD pin. While the transmitter is active, the data direction register control for Port D bit 1 is overridden and the line is forced to be an output.

RE — Receiver Enable

When the receiver enable bit is set, the receiver is enabled. When RE is clear, the receiver is disabled and all of the status bits associated with the receiver (RDRF, IDLE, OR, NF and FE) are inhibited. While the receiver is enabled, the data direction register control for Port D bit 0 is overridden and the line is forced to be an input.

RWU — Receiver Wake-up

When the receiver wake-up bit is set by the user software, it puts the receiver to sleep and enables the wake-up function. If the WAKE bit is cleared, RWU is cleared by the SCI logic after receiving 10 (M = 0) or 11 (M = 1) consecutive ones. If the WAKE bit is set, RWU is cleared by the SCI logic after receiving a data word whose MSB is set.

SBK — Send Break

If the send break bit is toggled set and cleared, the transmitter sends 10 (M = 0) or 11 (M = 1) zeros and then reverts to idle sending data. If SBK remains set, the transmitter will continually send whole blocks of zeros (sets of 10 or 11) until cleared. At the completion

of the break code, the transmitter sends at least one high bit to guarantee recognition of a valid start bit. If the transmitter is currently empty and idle, setting and clearing SBK is likely to queue two character times of break because the first break transfers almost immediately to the shift register and the second is then queued into the parallel transmit buffer.

7.8.4 Serial Communications Status Register (SCSR)

The serial communications status register (SCSR) provides inputs to the interrupt logic circuits for generation of the SCI system interrupt.

	7	6	5	4	3	2	1	0	
$102E	TDRE	TC	RDRF	IDLE	OR	NF	FE	0	SCSR
RESET:	1	1	0	0	0	0	0	0	

READ: Any time (used in auto clearing mechanism).

WRITE: Has no meaning or effect.

TDRE — Transmit Data Register Empty Flag

This bit is set when the byte in the transmit data register is transferred to the serial shift register. New data will not be transmitted unless the SCSR register is read before writing to the transmit data register. Reset sets this bit.

TC — Transmit Complete Flag

This bit is set to indicate that the SCI transmitter has no meaningful information to transmit (no data in shifter, no preamble, no break). When TC is set the serial line will go idle (continuous MARK). Reset sets this bit.

RDRF — Receive Data Register Full Flag

This bit is set when the contents of the receiver serial shift register is transferred to the receiver data register.

IDLE — Idle Line Detected Flag

This bit is set when a receiver idle line is detected (the receipt of a minimum of ten/eleven consecutive “1”s). This bit will not be set by the idle line condition when the RWU bit is set. Once cleared, IDLE will not be set again until after RDRF has been set, (until after the line has been active and becomes idle again).

OR — Overrun Error Flag

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

NF — Noise Error Flag

This bit is set if there is noise on a “valid” start bit, any of the data bits, or on the stop bit. The NF bit is set during the same cycle as the RDRF bit but does not get set in the case of an overrun (OR).

FE — Framing Error Flag

This bit is set when the word boundaries in the bit stream are not synchronized with the receiver bit counter (generated by the reception of a logic zero bit where a stop bit was expected). The FE bit reflects the status of the byte in the receive data register and the transfer from the receive shifter to the receive data register is inhibited in the case of overrun. The FE bit is set during the same cycle as the RDRF bit but does not get set in the case of an overrun (OR). The framing error flag inhibits further transfer of data into the receive data register until it is cleared.

7.8.5 Baud Rate Register (BAUD)

The baud rate register (BAUD) is used to set the bit rate for the SCI system. Normally this register is written once, during initialization, to set the baud rate for SCI communications. Both the receiver and the transmitter use the same baud rate which is derived from the MCU bus rate clock. A two stage divider is used to develop custom baud rates from normal MCU crystal frequencies so it is not necessary to use special baud rate crystal frequencies.

	7	6	5	4	3	2	1	0	
$102B	TCLR	0	SCP1	SCP0	RCKB	SCR2	SCR1	SCR0	BAUD
RESET:	0	0	0	0	0	U	U	U	

TCLR — Clear Baud Rate Counters (for test purposes only)

READ: Always returns 0.

WRITE: Only while SMOD = 1 (test or bootstrap mode)

This bit is disabled and remains low in any mode other than test or bootstrap mode. Reset clears this bit. While in test or bootstrap mode, setting this bit causes the baud rate counter chains to be reset. The logic one state of this bit is transitory and reads always return a logic zero. This control bit is intended only for factory testing of the MCU.

SCP1, SCP0 — Serial Prescaler Select bits

READ: Any time

WRITE: Any time

The E-clock is divided by the factors shown in Table 7-1. This prescaled output provides an input to a divider which is controlled by the SCI rate select bits (SCR2 – SCR0).

Table 7-1. First Prescaler Stage

SCP1	SCP0	Internal Processor Clock Divide By
0	0	1
0	1	3
1	0	4
1	1	13

SCR2, SCR1, SCR0 — SCI Rate Select bits

READ: Any time

WRITE: Any time

These three bits select the baud rates for both the transmitter and the receiver. The prescaler output described above is divided by the factors shown in Table 7-2.

Table 7-2. Second Prescaler Stage

SCR2	SCR1	SCR0	Prescaler Output Divide By
0	0	0	1
0	0	1	2
0	1	0	4
0	1	1	8
1	0	0	16
1	0	1	32
1	1	0	64
1	1	1	128

RCKB — SCI Receive Baud Rate Clock Test

READ: Always returns logic 0

WRITE: Only while SMOD = 1 (test or bootstrap mode)

This bit is disabled and remains low in any mode other than test or bootstrap modes. Reset clears this bit. While in test or bootstrap mode, this bit may be written but not read (reads always return a logic zero). Setting this bit enables a baud rate counter test mode where the exclusive-or of the receiver clock (16 times the baud rate) and the transmit clock (1 times the baud rate) is driven out the PD1/TXD pin. This control bit is intended only for factory testing of the MCU.

SECTION 8

SERIAL PERIPHERAL INTERFACE

This section contains a description of the serial peripheral interface (SPI).

8.1 OVERVIEW AND FEATURES

The SPI is a synchronous interface which allows several SPI microcontrollers or SPI-type peripherals to be interconnected. In a serial peripheral interface, separate wires (signals) are required for data and clock. In the SPI format, the clock is not included in the data stream and must be furnished as a separate signal. The MC68HC11G5 SPI system may be configured either as a master or as a slave.

Features include:

- Full-duplex, 3-wire synchronous transfers
- Master or slave operation
- 1.05 MHz (maximum) master bit frequency
- 2.1 MHz (maximum) slave bit frequency
- Four programmable master bit rates
- Programmable clock polarity and phase
- End-of-transmission interrupt flag
- Write collision flag protection
- Master-master mode fault protection
- Easy interface to simple expansion parts (PLLs, D/As, latches, display drivers, etc.)

8.2 SPI SIGNAL DESCRIPTIONS

The SPI interface consists of four Port D lines (MISO, MOSI, SCK, and $\overline{SS}$). These signals are discussed in the following paragraphs for both master mode and slave mode of operation.

Any SPI output line has to have its corresponding data direction register bit set. If this bit is clear, the line is disconnected from the SPI logic and becomes a general-purpose input line. Any SPI input line is forced to act as an input regardless of the state of the corresponding data direction register bit.

8.2.1 Master In Slave Out (MISO)

The MISO line is configured as an input in a master device and as an output in a slave device. It is one of the two lines that transfer serial data in one direction, with the most significant bit sent first. The MISO line of a slave device is placed in the high-impedance state if the slave is not selected.

8.2.2 Master Out Slave In (MOSI)

The MOSI line is configured as an output in a master device and as an input in a slave device. It is one of the two lines that transfer serial data in one direction with the most significant bit sent first.

8.2.3 Serial Clock (SCK)

The serial clock is used to synchronize data movement both in and out of the device through its MOSI and MISO lines. The master and slave devices are capable of exchanging a byte of information during a sequence of eight clock cycles. Since SCK is generated by the master device, this line becomes an input on a slave device.

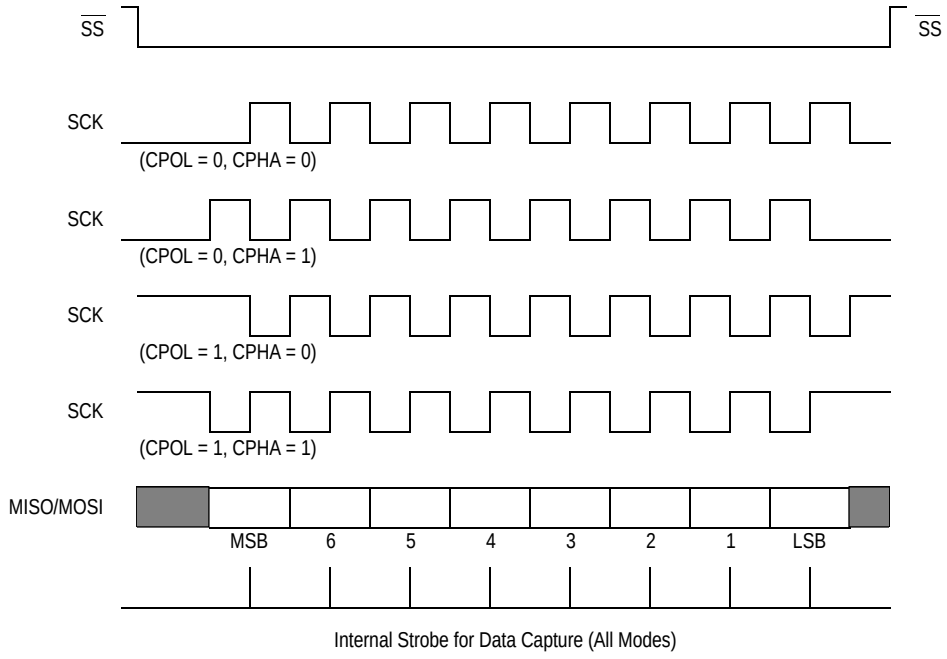


Figure 8-1. Data Clock Timing Diagram

As shown in Figure 8-1, four different timing relationships may be selected by control bits CPOL and CPHA in the serial peripheral control register (SPCR). Both master and slave devices must operate with the same timing. The master device always places data on the MOSI line a half cycle before the clock edge (SCK), in order for the slave device to latch the data.

Two bits (SPR0 and SPR1) in the SPCR of the master device select the clock rate. In a slave device, SPR0 and SPR1 have no effect on the operation of the SPI.

8.2.4 Slave Select ($\overline{SS}$)

The slave select ($\overline{SS}$) input line is used to select a slave device. It must be in the active low state prior to data transactions and must stay low for the duration of the transaction.

The $\overline{SS}$ line on the master must be tied high. If it goes low, a mode fault error flag (MODF) is set in the serial peripheral status register (SPSR). The $\overline{SS}$ pin can be selected to be a general-purpose output by writing a one in bit 5 of the Port D data direction register, thus disabling the mode fault circuit. The other three SPI lines are dedicated to the SPI whenever the SPI is on.

When CPHA = 0, the shift clock is the logical OR of $\overline{SS}$ and SCK. In this clock phase mode, $\overline{SS}$ must go high between successive characters in an SPI message. When CPHA = 1, $\overline{SS}$ may be left low for several SPI characters. If there is only one SPI slave MCU, its $\overline{SS}$ line may be tied to V_{SS} provided CPHA = 1 clock modes are used.

Figure 8-2 shows a block diagram of the serial peripheral interface circuitry. When a master device transmits data to a slave device via the MOSI line, the slave device responds by sending data to the master device via the master's MISO line. This implies full duplex transmission with both data out and data in synchronized to the same clock signal. Thus, the byte transmitted is replaced by the byte received and eliminates the need for separate transmitter-empty and receiver-full status bits. A single status bit (SPIF) is used to signify that the I/O operation has been completed.

The SPI is double buffered on read, but not on write. If a write is performed during data transfer, the transfer is not interrupted, and the write will be unsuccessful. This condition will cause the write collision status bit (WCOL) in the SPSR to be set. After a data byte is shifted, the SPIF flag in the SPSR is set.

In master mode, the SCK pin is an output. It idles high or low, depending on the SPOL bit in the SPCR, until data is written to the shift register, at which point eight clocks are generated to shift the eight bits of data, after which SCK goes idle again.

In slave mode, the slave start logic receives a logic low on the $\overline{SS}$ pin and a clock input at the SCK pin. Thus, the slave is synchronized to the master. Data from the master is received serially via the slave MOSI line and is loaded into the 8-bit shift register. The data is then transferred, in parallel, from the 8-bit shift register to the read buffer. During a write cycle, data is written into the shift register, then the slave waits for a clock train from the master to shift the data out on the slave's MISO line.

Figure 8-3 illustrates the MOSI, MISO and $\overline{SS}$ master-slave interconnections.

8.3 FUNCTIONAL DESCRIPTION

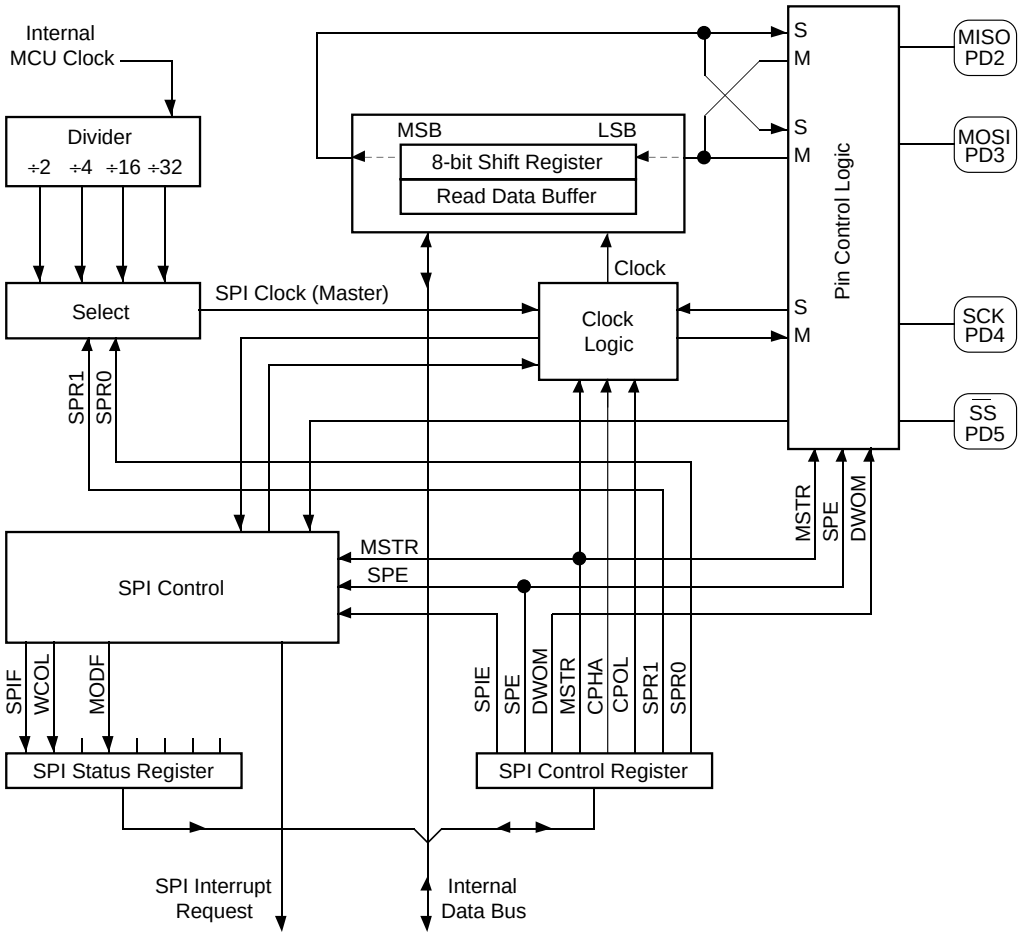


Figure 8-2. Serial Peripheral Interface Block Diagram

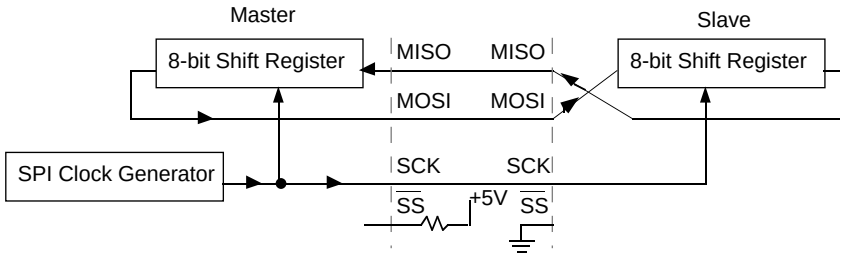


Figure 8-3. Serial Peripheral Interface Master-Slave Interconnection

Due to data direction register control of SPI outputs and the Port D wire-OR mode (DWOM) option, the SPI system can be configured in a variety of ways. Systems with a single bidirectional data path rather than separate MISO and MOSI paths can be accommodated. Since MC68HC11G5 slaves can selectively disable their MISO output, a broadcast message protocol is also possible.

8.4 SPI REGISTERS

There are three registers in the serial peripheral interface which provide control, status and data storage functions. These registers are called the serial peripheral control register (SPCR), the serial peripheral status register (SPSR) and the serial peripheral data I/O register (SPDAT).

8.4.1 Control Register (SPCR)

	7	6	5	4	3	2	1	0	
$1028	SPIE	SPE	DWOM	MSTR	CPOL	CPHA	SPR1	SPR0	SPCR
RESET:	0	0	0	0	0	1	U	U	

READ: Any time

WRITE: Any time.

SPIE — SPI Interrupt Enable

0 – SPI interrupts disabled.

1 – SPI interrupts enabled.

When this bit is set to one, a hardware interrupt sequence is requested 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 CC Register is set.

SPE — SPI System Enable

0 – SPI system off.

1 – SPI system on.

When the SPE bit is set the Port D bit 2, 3, 4, and 5 pins are dedicated to the SPI function. If the SPI is in master mode and DDRD bit 5 is set, then the Port D bit 5 pin becomes a general purpose output instead of the $\overline{SS}$ input.

DWOM — Port D Wired-OR Mode

0 – Port D output buffers operate normally (CMOS outputs).

1 – Port D output buffers operate as open-drain outputs.

DWOM affects all six Port D pins together.

MSTR — Master/Slave Mode Select

0 – Slave mode.

1 – Master mode.

CPOL — Clock Polarity

When the clock polarity bit is cleared and data is not being transferred, a steady state low value is produced at the SCK pin of the master device. Conversely, if this bit is set, the SCK pin will idle high. This bit is also used in conjunction with the clock phase control bit to produce the desired clock-data relationship between master and slave. See Figure 8-1.

CPHA — Clock Phase

The clock phase bit, in conjunction with the CPOL bit, controls the clock-data relationship between master and slave. The CPOL bit can be thought of simply as inserting an inverter in series with the SCK line. The CPHA bit selects one of two fundamentally different clocking protocols. When CPHA = 0, the shift clock is the logical OR of SCK and $\overline{SS}$. As soon as $\overline{SS}$ goes low, the transaction begins and the first edge on SCK invokes the first data sample. When CPHA = 1, the $\overline{SS}$ pin may be thought of as a simple output enable control. Refer to Figure 8-1.

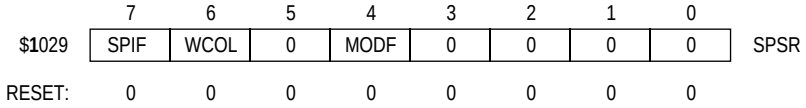
SPR1, SPR0 — SPI Clock (SCK) Rate Select Bits

If the device is a master, the two serial peripheral rate bits select one of four division ratios of the E-clock to be used as SCK (see Table 8-1). These bits have no effect in slave mode.

Table 8-1. SPI Rate Selection

SPR1	SPR0	Internal Processor Clock Divide By
0	0	2
0	1	4
1	0	16
1	1	32

8.4.2 Status Register (SPSR)



	7	6	5	4	3	2	1	0	
$1029	SPIF	WCOL	0	MODF	0	0	0	0	SPSR
RESET:	0	0	0	0	0	0	0	0	

READ: Any time.

WRITE: Has no meaning or effect.

SPIF — SPI Interrupt Request Flag

The serial peripheral data transfer flag bit is set after the eighth SCK cycle in a data transfer and it is cleared by reading the SPSR register (with SPIF set) followed by reading from or writing to the SPI Data Register (SPDAT).

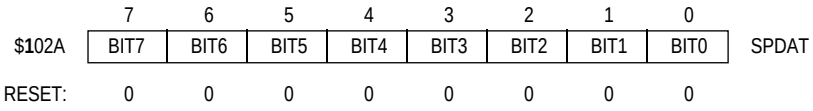
WCOL — Write Collision

The write collision bit is used to indicate that a serial transfer was in progress when the MCU tried to write new data into the SPDAT data register. The MCU write is disabled to avoid writing over the data being transmitted. No interrupt is generated because the error status flag can be read upon completion of the transfer that was in progress at the time of the error. This flag is automatically cleared by a read of the SPSR (with WCOL set) followed by an access (read or write) to the SPDAT register.

MODF — SPI Mode Error Interrupt Status Flag

This bit is set automatically by SPI hardware if the MSTR control bit is set to one and the $\overline{SS}$ input pin goes low. This condition is not permitted in normal operation. In the special case where DDRD bit 5 is set to one, the Port D bit 5 pin is a general purpose output pin rather than being dedicated as the slave select input for the SPI system. In this special case the mode error function is inhibited and MODF remains at zero. This flag is automatically cleared by a read of the SPSR (with MODF set) followed by a write to the SPCR register.

8.4.3 Data I/O Register (SPDAT)



	7	6	5	4	3	2	1	0	
$102A	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0	SPDAT
RESET:	0	0	0	0	0	0	0	0	

READ: Any time (normally only after SPIF flag set)

WRITE: Any time (see WCOL write collision flag).

RESET: Does not affect this register.

The serial peripheral data I/O register is used to transmit and receive data on the serial bus. Only a write to this register will initiate transmission/reception of another byte, and this will only occur in

the master device. At the completion or transmitting a byte of data, the SPIF status bit is set in both the master and slave devices.

When the user reads the serial peripheral data I/O register, a buffer is actually being read. The first SPIF must be cleared by the time a second transfer of data from the shift register to the read buffer is initiated or an overrun condition will exist. In cases of overrun, the byte which causes the overrun is lost.

A write to the serial peripheral data I/O register is not buffered and places data directly into the shift register for transmission.

SECTION 9

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. Eight of the channels are connected to pins on the MC68HC11G5, four are unused and the remaining four channels are dedicated to internal reference points or test functions. 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, the control logic allows conversions to be performed on a single selected channel, multiple times or consecutively on a selected group of four channels. In addition, the control logic allows for converting all eight channels and either stopping or converting continuously.

Two dedicated lines (V_{rl} and V_{rh}) 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 AD conversion.

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_{rh} 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.

Two control bits in the OPTION register control the basic configuration of the A/D system. The A to 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.

9.1 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.

9.2 CHANNEL ASSIGNMENTS

A multiplexer allows the single A/D converter to select one of sixteen analog signals. Eight of these channels are supported on the Port E input pins. Of the eight other channels, four are reserved for future use and four are for internal reference points and testing purposes. Table 9-1 shows the signals selected by the channel select bits (CD, CC, CB, CA) in the ADCTL register. All “reserved” channels are connected to V_{rl}.

Table 9-1. Channel Assignments

CONV8 = 0					Result in register if	
CD	**CC**	**CB**	**CA**	**Channel Signal**	**MULT = 1**	**MULT = 0**
0	0	0	0	AN1 (PE0)	ADR5	ADR5 – 8
0	0	0	1	AN2 (PE1)	ADR6	ADR5 – 8
0	0	1	0	AN3 (PE2)	ADR7	ADR5 – 8
0	0	1	1	AN4 (PE3)	ADR8	ADR5 – 8
0	1	0	0	AN5 (PE4)	ADR5	ADR5 – 8
0	1	0	1	AN6 (PE5)	ADR6	ADR5 – 8
0	1	1	0	AN7 (PE6)	ADR7	ADR5 – 8
0	1	1	1	AN8 (PE7)	ADR8	ADR5 – 8
1	0	X	X	reserved		
1	1	0	0	V_{rh}	ADR5	ADR5 – 8
1	1	0	1	V_{rl}	ADR6	ADR5 – 8
1	1	1	0	$V_{rh}/2$	ADR7	ADR5 – 8
1	1	1	1	Test (reserved)	ADR8	ADR5 – 8

Table 9-1. Channel Assignments (continued)

CONV8 = 1					Result in register if	
CD	CC	CB	CA	Channel Signal	MULT = 1	MULT = 0
0	0	0	0	AN1 (PE0)	ADR1	ADR1 – 8
0	0	0	1	AN2 (PE1)	ADR2	ADR1 – 8
0	0	1	0	AN3 (PE2)	ADR3	ADR1 – 8
0	0	1	1	AN4 (PE3)	ADR4	ADR1 – 8
0	1	0	0	AN5 (PE4)	ADR5	ADR1 – 8
0	1	0	1	AN6 (PE5)	ADR6	ADR1 – 8
0	1	1	0	AN7 (PE6)	ADR7	ADR1 – 8
0	1	1	1	AN8 (PE7)	ADR8	ADR1 – 8
1	0	X	X	reserved		
1	1	0	0	V_{rh}	ADR5	ADR1 – 8
1	1	0	1	V_{rl}	ADR6	ADR1 – 8
1	1	1	0	$V_{rh}/2$	ADR7	ADR1 – 8
1	1	1	1	Test (reserved)	ADR8	ADR1 – 8

9.3 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.

9.3.1 4-Conversion, Single Scan

Whichever Port E bit 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.

9.3.2 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.

9.3.3 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.

9.3.4 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.

9.4 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 either Port E bits 0 – 3 or Port E bits 4 – 7 are selected. (In this multiple channel mode, only the two most significant bits of the channel address (CD and CC) are decoded.) In the second two variations, the CONV8 bit is set and all eight channels are converted. The state of the SCAN bit determines whether continuous or single scanning is selected.

9.4.1 4-Channel Single Scan

Either Port E bits 0 – 3 are selected or Port E bits 4 – 7 are selected. The first result is stored in the ADR5 result register and the fourth result is 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.

9.4.2 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.

9.4.3 8-Channel Single Scan

When CONV8 is set and MULT is set, all eight channels are converted. Each of the channels is converted and the result is placed in a separate result register. Port E bit 0 uses result register ADR1, Port E bit 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.

9.4.4 8-Channel Continuous Scan

Conversions continue to be performed on all 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.

9.5 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.

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.

9.6 OPERATION IN STOP AND WAIT MODES

If a conversion sequence is still in process when the MC68HC11G5 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 MC68HC11G5 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 MC68HC11G5 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.

9.7 REGISTERS

9.7.1 A/D Control and Status Register (ADCTL)

	7	6	5	4	3	2	1	0	
$1030	CCF	CONV8	SCAN	MULT	CD	CC	CB	CA	ADCTL
RESET:	Undefined								

READ: Any time.

WRITE: Any time except writes always clear bit 7.

RESET: Indeterminate.

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 to zero, any current conversion is aborted and a new conversion sequence is started.

CONV8 — Convert 8/Convert 4 Select Bit

0 – Convert 4 channels or one channel 4 times (uses 4 result registers).

1 – Convert 8 channels or one channel 8 times, (uses all 8 result registers).

SCAN — Continuous Scan Control

0 – Perform selected number of conversions (4 or 8) and stop.

1 – Convert continuously.

MULT — Multiple Channel/Single Channel Control

0 – Convert single channel selected.

1 – Convert the selected number of channels (4 or 8).

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 9-1 for a list of the A/D channel assignments.)

9.7.2 Result Registers (ADR1 – ADR8)

	7	6	5	4	3	2	1	0	
$1040	BIT15	BIT14	BIT13	BIT12	BIT11	BIT10	BIT9	BIT8	ADR1
$1041	BIT7	BIT6	0	0	0	0	0	0	
$1042	BIT15	BIT14	BIT13	BIT12	BIT11	BIT10	BIT9	BIT8	ADR2
$1043	BIT7	BIT6	0	0	0	0	0	0	
$1044	BIT15	BIT14	BIT13	BIT12	BIT11	BIT10	BIT9	BIT8	ADR3
$1045	BIT7	BIT6	0	0	0	0	0	0	
$1046	BIT15	BIT14	BIT13	BIT12	BIT11	BIT10	BIT9	BIT8	ADR4
$1047	BIT7	BIT6	0	0	0	0	0	0	
$1048	BIT15	BIT14	BIT13	BIT12	BIT11	BIT10	BIT9	BIT8	ADR5
$1049	BIT7	BIT6	0	0	0	0	0	0	
$104A	BIT15	BIT14	BIT13	BIT12	BIT11	BIT10	BIT9	BIT8	ADR6
$104B	BIT7	BIT6	0	0	0	0	0	0	
$104C	BIT15	BIT14	BIT13	BIT12	BIT11	BIT10	BIT9	BIT8	ADR7
$104D	BIT7	BIT6	0	0	0	0	0	0	
$104E	BIT15	BIT14	BIT13	BIT12	BIT11	BIT10	BIT9	BIT8	ADR8
$104F	BIT7	BIT6	0	0	0	0	0	0	

RESET: Undefined

READ: Any time.

WRITE: Has no meaning or effect.

RESET: Indeterminate.

The eight 10-bit result registers are read-only. In each result register, the 8 high order bits are in one address location and the remaining 2 low order bits are in bit locations 6 and 7 of the following address. The 6 unused bits will always read as zeros. This allows a double byte read to be performed without having to adjust the result.

This page intentionally left blank

SECTION 10

PULSE WIDTH MODULATION TIMER

10.1 GENERAL OVERVIEW

The PWM module provides up to four 8-bit pulse width modulated waveforms on the Port H pins. Channel pairs may be concatenated to create 16-bit PWM outputs. Three clock sources (A, B and S) give the PWM a wide range of frequencies. Figure 10-1 shows the block diagram of the PWM module.

Each channel also has a separate 8-bit counter register (PWCNTx), period register (PWPERx), and duty cycle register (PWDTYx). Resetting the counter and starting the waveform output is controlled by the occurrence of a match between the period register and the value in the counter. Counters may also be reset by writing to them. The duty register changes the state of the output during the period to control the duty cycle of the waveform. The period and duty cycle registers are double buffered so that, if they are changed while the channel is enabled, the change will not take effect until the counter rolls over or the channel is disabled. A new period or duty cycle can be forced by writing to the period or duty cycle register and then writing to the counter.

Four control registers are used to configure the PWM outputs – PWCLK, PWPOL, PWSCAL, and PWEN. The PCKAx and PCKBx bits in the PWCLK register select the prescale values for the PWM clock sources. PWCLK also contains the CON34 and CON12 bits which enable the 16-bit PWM functions. The PWPOL register determines each channel's polarity (PPOLx bits) and selects the clock source for each channel (PCLKx bits). The PWSCAL register can be programmed with a prescale value which is used to derive a scaled clock based on the A clock source. The PWENx bits in the PWEN register enable or disable the PWM channels.

With PWMs configured for 8-bit mode and E = 2 MHz, PWM signals can be produced from 20 kHz (1% duty cycle resolution) to less than 5 Hz (approximately 0.4% duty cycle resolution). By configuring the PWMs for 16-bit mode with E = 2 MHz, PWM periods greater than one minute can be achieved.

In 16-bit mode, duty cycle resolution of almost 15 parts per million can be achieved (at a PWM frequency of about 30 Hz). In the same system, a PWM frequency of 1 kHz would correspond to a duty cycle resolution of 0.05%.

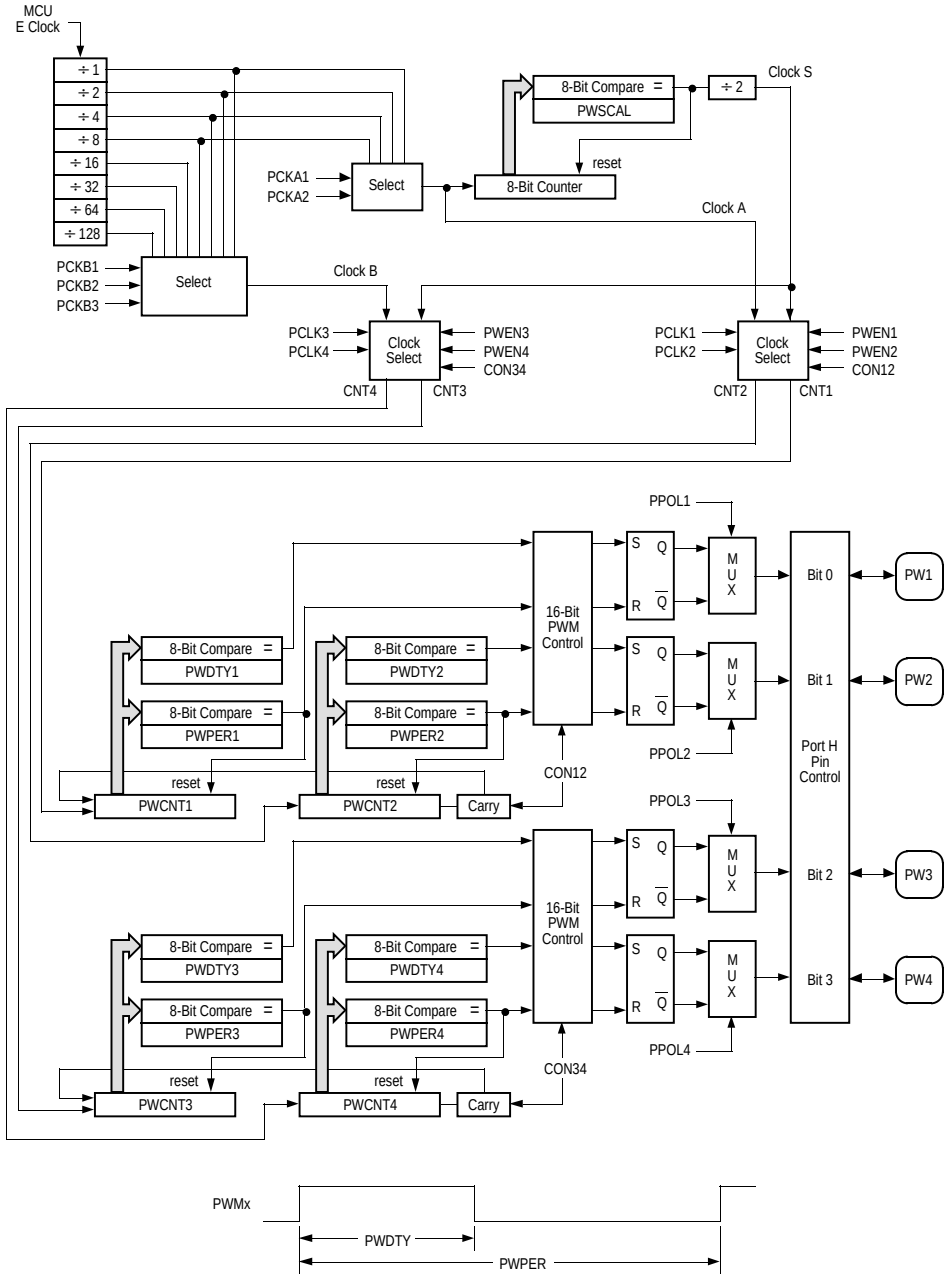


Figure 10-1. PWM Timer Module Block Diagram

10.2 CLOCK SELECTION

There are three available clocks (A, B and S). 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 block diagram of Figure 10-1 depicts the three different clocks and how the scaled clock is created.

The scaled clock (S) uses clock A as an input and divides it with a reloadable counter. This counter is compared with a user programmable scale value (in the PWSCAL register). When they match, a pulse is output and the 8-bit counter is reset. The output signal from this circuit is then divided by two to give clock S. The rates available for clock S, therefore, are software selectable to be clock A divided by 2 down to clock A divided by 512 in increments of 2.

Each PWM timer channel can be driven by one of two clocks, selected in software: Channels 1 and 2 can use clock A or clock S; Channels 3 and 4 can use clock B or clock S.

Writing to PWSCAL causes the 8-bit counter to be reset to $00. Otherwise, when changing from a low rate to a higher rate, it would be possible for the counter to miss the new value and have to go all the way to $FF and wrap back around before counting at the proper rate. Forcing the counter to $00 every time it is written prevents this.

10.3 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. Concatenation of channels 3 and 4 is controlled by the CON34 bit; similarly, channels 1 and 2 are concatenated using the CON12 bit.

When channels 3 and 4 are concatenated, channel 3 registers become the high order bytes of the double byte channel. Reading the counter high order byte causes the low order byte to be latched for one cycle, to guarantee that double byte reads will be accurate. Writing to the low byte of the counter (channel 4) causes the entire counter to be reset. Writing to the upper byte of the counter has no effect.

Similarly, when channels 1 and 2 are concatenated, channel 1 registers become the high order bytes of the double byte channel.

The 16-bit duty register and period register are obtained by concatenating the two 8-bit duty registers and period registers, respectively. Writing to these registers takes twice as many write cycles as for the 8-bit PWM. The 16-bit PWM circuit has secondary buffers internally, and the user should write to these 16-bit registers using 16-bit write instructions or high byte to low byte sequential write instructions. During the two-byte write sequence, the secondary buffer will not be loaded from the 16-bit register. The 16-bit duty and period registers can be written to at any time; the written value will take effect from the next cycle of the PWM period.

10.4 BOUNDARY CASES

The following boundary conditions cause these results:

If PWDTYx = $00, PWPERx > $00, PPOLx = 0;	the output is always high.
If PWDTYx = $00, PWPERx > $00, PPOLx = 1;	the output is always low.
If PWDTYx = or > PWPERx, PPOLx = 0;	the output is always low.
If PWDTYx = or > PWPERx, PPOLx = 1;	the output is always high.
If PWPERx = $00 & PPOLx = 0;	output x is always low.
If PWPERx = $00 & PPOLx = 1;	output x is always high.

10.5 PWM REGISTERS

10.5.1 Counter Registers (PWCNTX)

Each channel has its own counter. Each counter may be read at any time without affecting the count or the operation of the PWM channel. Writing to a counter causes the counter to be reset to $00. Generally, writing to a counter is done before the counter is enabled. Writing to a counter can also be done while it is enabled (counting), but this may cause a truncated PWM period.

	7	6	5	4	3	2	1	0	
$1064	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0	PWCNT1
$1065	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0	PWCNT2
$1066	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0	PWCNT3
$1067	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0	PWCNT4

READ: Any time.

WRITE: Forces value to $00.

RESET: $00.

When a channel becomes enabled (i.e. when PWENx is written from 0 to 1), the associated PWM counter starts to count from the value in this PWCNTx register, using whichever clock has been selected for that channel.

10.5.2 Period Registers (PWPERX)

There is one period register for each channel. The value in this register determines the period of the associated PWM timer channel.

In terms of the internal PWM circuitry, this register is connected to a buffer which compares with the counter register directly. The period value in this register 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 next PWM timer cycle. Reading this register returns the most recent value written.

	7	6	5	4	3	2	1	0	
$1068	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0	PWPER1
$1069	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0	PWPER2
$106A	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0	PWPER3
$106B	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0	PWPER4

READ: Any time.

WRITE: Any time.

RESET: $FF.

10.5.3 Duty Registers (PWDTYx)

There is one duty register for each channel. The value in this register determines the duty cycle of the associated PWM timer channel.

In terms of the internal PWM circuitry, this register is connected to a buffer which compares with the counter directly. The duty value in this register is loaded into the second buffer when the counter is cleared by previous period termination or the counter write operation. This register can be written at any time, and the written value will take effect from the next PWM timer cycle. Reading this register returns the most recent value written.

Note: If the duty register is greater than or equal to the value in the period register there will be no duty change in state. In addition, if the duty register is set to $00 the output will always be in the state which would normally be the state changed to at the duty change of state. Refer to the “Boundary Cases” section for more information on this.

	7	6	5	4	3	2	1	0	
$106C	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0	PWDTY1
$106D	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0	PWDTY2
$106E	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0	PWDTY3
$106F	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0	PWDTY4

READ: Any time.

WRITE: Any time.

RESET: $FF.

10.5.4 Clock Select Register (PWCLK)

	7	6	5	4	3	2	1	0	
$1060	CON34	CON12	PCKA2	PCKA1	0	PCKB3	PCKB2	PCKB1	PWCLK
RESET:	0	0	0	0	0	0	0	0	

READ: Any time.

WRITE: Any time.

RESET: $00.

CON34 — Concatenate channels 3 and 4

1 – Channels 3 and 4 are concatenated to create one 16-bit PWM channel. (Channel 3 becomes the high order byte and channel 4 becomes the low order byte. The channel 4 output is used as the output for this 16-bit PWM (bit 3 of Port H)).

0 – Channels 3 and 4 are separate 8-bit PWM channels.

CON12 — Concatenate channels 1 and 2

1 – Channels 1 and 2 are concatenated to create one 16-bit PWM channel. (Channel 1 becomes the high order byte and channel 2 becomes the low order byte. The channel 2 output is used as the output for this 16-bit PWM (bit 1 of Port H)).

0 – Channels 1 and 2 are separate 8-bit PWM channels.

PCKA2, PCKA1 — Prescaler for clock A

Clock A is one of two clock sources which may be used for channels 1 & 2. These two bits determine the rate of clock A.

PCKA2	PCKA1	Value of Clock A
0	0	E
0	1	E/2
1	0	E/4
1	1	E/8

PCKB3, PCKB2, PCKB1 — Prescaler for clock B

Clock B is one of two clock sources which may be used for channels 3 & 4. These three bits determine the rate of clock B.

PCKB3	PCKB2	PCKB1	Value of 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

10.5.5 Polarity Select Register (PWPOL)

Each channel has a polarity bit (PPOLx) to start the cycle with a high signal or with a low signal. This is shown on the block diagram as a multiplex select 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.

	7	6	5	4	3	2	1	0	
$1061	PCLK4	PCLK3	PCLK2	PCLK1	PPOL4	PPOL3	PPOL2	PPOL1	PWPOL
RESET:	0	0	0	0	0	0	0	0	

READ: Any time.

WRITE: Any time.

RESET: $00

PCLK4 — Pulse Width Channel 4 Clock Select

1 – The clock source for PWM channel 4 is Clock S. (Clock S equals Clock A divided by two times the value (plus one) in the PWSCAL register.)

0 – Clock B is the clock source for PWM channel 4.

PCLK3 — Pulse Width Channel 3 Clock Select

1 – The clock source for PWM channel 3 is Clock S.

0 – Clock B is the clock source for PWM channel 3.

PCLK2 — Pulse Width Channel 2 Clock Select

1 – The clock source for PWM channel 2 is Clock S.

0 – Clock A is the clock source for PWM channel 2.

PCLK1 — Pulse Width Channel 1 Clock Select

1 – The clock source for PWM channel 1 is Clock S.

0 – Clock A is the clock source for PWM channel 1.

Note: While register bits PCLK1 – 4 may be written at any time, if a clock select is changed while a PWM signal is being generated, a truncated pulse may occur during the transition.

PPOL4 — Pulse Width Channel 4 Polarity

1 – PWM channel 4 output is high at the beginning of the clock cycle, then goes low when the duty count is reached.

0 – PWM channel 4 output is low at the beginning of the clock cycle, then goes high when the duty count is reached.

PPOL3 — Pulse Width Channel 3 Polarity

1 – PWM channel 3 output is high at the beginning of the clock cycle, then goes low when the duty count is reached.

0 – PWM channel 3 output is low at the beginning of the clock cycle, then goes high when the duty count is reached.

PPOL2 — Pulse Width Channel 2 Polarity

1 – PWM channel 2 output is high at the beginning of the clock cycle, then goes low when the duty count is reached.

0 – PWM channel 2 output is low at the beginning of the clock cycle, then goes high when the duty count is reached.

PPOL1 — Pulse Width Channel 1 Polarity

1 – PWM channel 1 output is high at the beginning of the clock cycle, then goes low when the duty count is reached.

0 – PWM channel 1 output is low at the beginning of the clock cycle, then goes high when the duty count is reached.

10.5.6 Scale Register (PWSCAL)

	7	6	5	4	3	2	1	0	
$1062	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0	PWSCAL
RESET:	0	0	0	0	0	0	0	0	

READ: Any time.

WRITE: Any time (causes clock counter to be reset to $00).

RESET: $00

Each of the PWM channels can select clock S (scaled) as its input clock. Clock S may be selected for channel x by writing a one to the control bit PCLKx. Clock S is generated by taking clock A, dividing it by the value in this PWSCAL register and dividing that by two. When PWSCAL = $00, clock A is divided by 256 then divided by two to generate clock S. In test mode, the PWSCAL register can be used to read the value of clock S (scaled) if the TPWSL bit in the TEST1 register is set.

10.5.7 Enable Register (PWEN)

Each timer has an enable bit (PWENx) to start its waveform output. Writing any of these PWENx bits to one causes the associated Port H 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 the I/O state. On the front end of the PWM timer the clock is enabled to the PWM circuit by the PWENx enable bit being high. A synchronizing circuit guarantees that the clock will only be enabled or disabled at an edge.

	7	6	5	4	3	2	1	0	
$1063	0	0	0	0	PWEN4	PWEN3	PWEN2	PWEN1	PWEN
RESET:	0	0	0	0	0	0	0	0	

READ: Any time.

WRITE: Any time.

RESET: $00.

PWEN4 — Pulse Width Channel 4 Enable

1 – PWM channel 4 is enabled. The pulse modulated signal becomes available at Port H bit 3 when its clock source begins its next cycle.

0 – PWM channel 4 is disabled.

PWEN3 — Pulse Width Channel 3 Enable

1 – PWM channel 3 is enabled. The pulse modulated signal becomes available at Port H bit 2 when its clock source begins its next cycle.

0 – PWM channel 3 is disabled.

PWEN2 — Pulse Width Channel 2 Enable

1 – PWM channel 2 is enabled. The pulse modulated signal becomes available at Port H bit 1 when its clock source begins its next cycle.

0 – PWM channel 2 is disabled.

PWEN1 — Pulse Width Channel 1 Enable

1 – PWM channel 1 is enabled. The pulse modulated signal becomes available at Port H bit 0 when its clock source begins its next cycle.

0 – PWM channel 1 is disabled.

SECTION 11

EVENT COUNTER

11.1 INTRODUCTION

The event counter consists of two distinct functional blocks. One block functions as an 8-bit pulse accumulator unit (PA), the other as an 8-bit pulse width modulation unit (PWM). Each unit has its own 8-bit counter and two 8-bit compare registers.

Four software selectable operating modes provide different configurations of the PA and PWM, their interconnection and their combined operation. (The operating mode is selected using the event counter mode select bits, EVMDA and EVMDB in the event counter lock register (EVCLK)). All four modes provide programmable PWM period, duty cycle and polarity. Note that these PWM circuits differ from the main PWM section in that these can be driven by an external signal. A simplified block diagram of each mode is shown in Figure 11-1.

In mode 0, the PA unit can modify the PWM output by adding a programmable offset to the input signal and by controlling the clearing of the PWM unit, thereby controlling the period of the output signal.

In mode 1, the event counter operates as separate PWM and PA units. The two units are not interconnected and operate completely independently of each other, providing standard programmable PWM and PA functions.

In mode 2, the PA unit counts the PWM output signal periods, which may be of varying duration and modulation.

In mode 3, the PA unit acts as a programmable 8-bit (1 – 256 clock) prescaler for the PWM unit.

The 8-bit counters in the PWM and PA units can be driven by an external clock or by a derivative of the E-clock (E or scaled E), generated by the on-chip time-base counter. The clock source is controlled by two input selectors (INPUT1 and INPUT2). Depending on the mode of operation selected, the input selectors are used to determine the following:

1) Clock source,
2) Active edge of the external signal (which may be used as a clock),
3) The gated input of the internal clock,
4) Counter clear

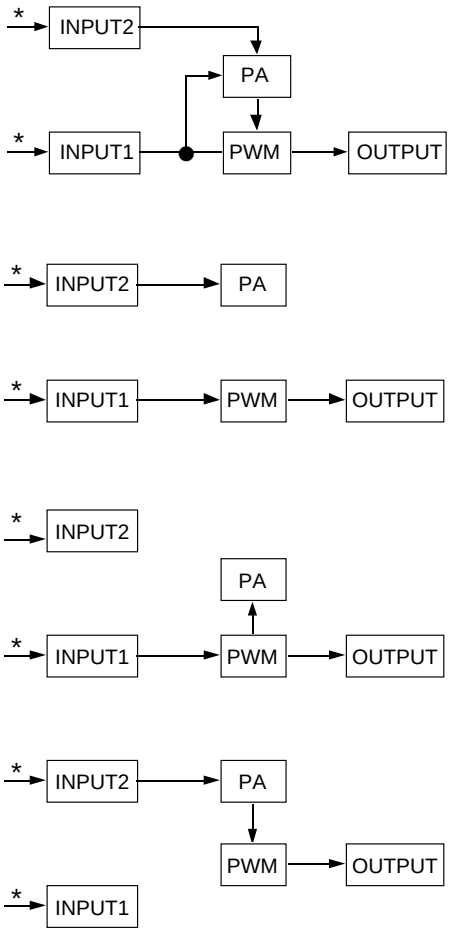
One output selector is used to control the polarity of the PWM output signal.

Mode 0: Pulse Accumulator modifies PWM output by adding offset and controlling the clearing of the PWM

Mode 1: PA and PWM are completely independent

Mode 2: PA Counts the PWM periods (which may be of variable duration and modulation)

Mode 4: PA acts as programmable 8-bit prescaler for PWM unit



* INPUT1 and INPUT2 can be driven from the E-clock, a second value of the E-clock or from external signals on the EVI2 and EVI1 input pins. When EVI2 or EVI1 is not being used for this purpose they may be used as general I/O pins (PH4 and PH5).

Figure 11-1. Event Counter Operating Modes

Each unit can generate an interrupt signal (referred to as EVENT1 or EVENT2 in the following discussion) on a counter match with one of its two compare registers.

The PWM unit is comprised of counter 1 (EVCNT1), two compare registers (ECMP1A and ECMP1B), an input selector (INPUT1) and an output selector. The PA unit is comprised of Counter 2 (EVCNT2), two compare registers (ECMP2A and ECMP2B) and an input selector (INPUT2).

11.2 MODE 0: 8-BIT PWM WITH SELECTABLE PHASE SHIFT

In mode 0, the event counter operates as an 8-bit PWM unit with an 8-bit phase shifter. If mode 0 is selected (EVMDB = 0, EVMDA = 0), then the elements of the event counter are configured as shown in Figure 11-2.

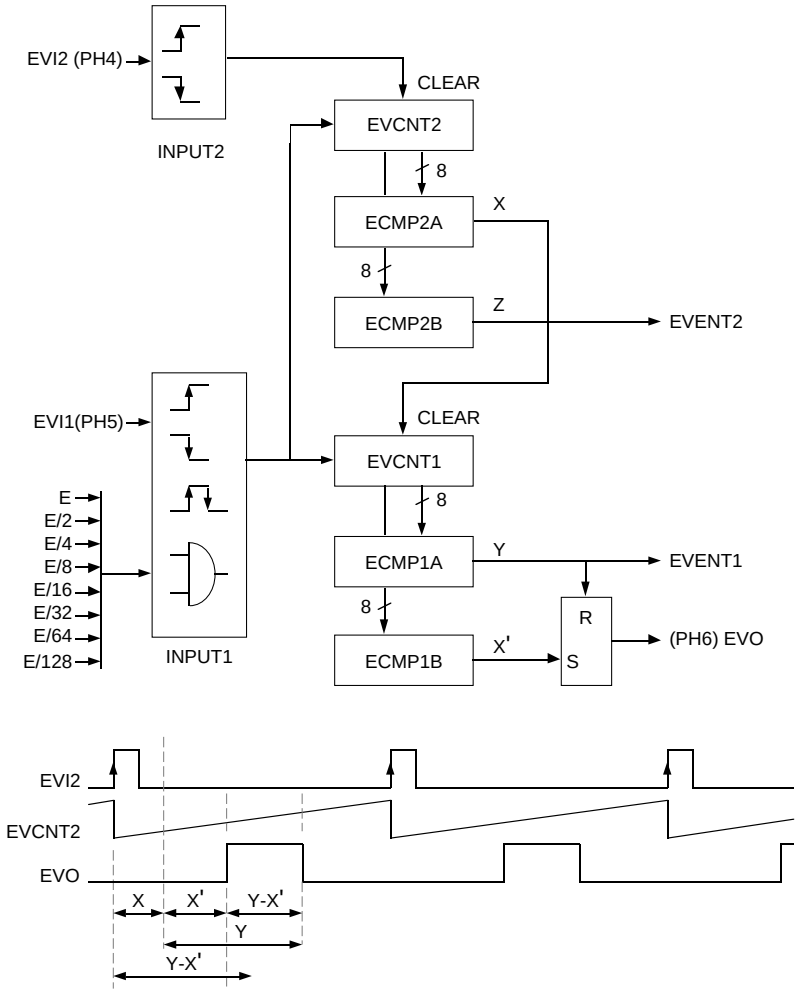


Figure 11-2. 8-bit PWM with Selectable Phase Shift (Mode 0)

11.2.1 Operation of PWM and PA Unit in Mode 0

The period of the modulated output is controlled by an external signal applied to the EVI2 pin. The active edge of this input signal is selected via the INPUT2 selector. This input signal clears EVCNT2 in the PA unit and starts the 8-bit phase shift sequence.

The two counter registers (EVCNT1 and EVCNT2) are clocked by the same signal via the INPUT1 selector. This clock signal can be the E-clock, a scaled E-clock or an external signal applied to the EVI1 pin. If the clock signal is the E-clock or a derivative, an external gating signal can be applied to EVI1. If EVI1 is not used for clock or gate, it can be used as a normal I/O pin (PH5).

The 8-bit phase shift value (X) is stored in ECMP2A (in the PA unit). When the count in EVCNT2 reaches this value, the successful compare causes a clearing signal to be applied to EVCNT1 in the PWM unit and starts the PWM sequence.

The value of the output pulse width (Y) is stored in ECMP1A, however the start time and the duration of this pulse can be modified by a skew value (X') which is stored in ECMP1B. If no skew is included, i.e. ECMP1B is set to zero, then a compare occurs when EVCNT1 is cleared, and the output signal changes state immediately after the initial phase shift period X introduced by the PA unit. The effect of storing a skew value (X') in ECMP1B is to delay the leading edge of the PWM output pulse by the phase shift plus the skew value (X+X').

The trailing edge of the output pulse is generated when there is a match between EVCNT1 and ECMP1A (X + Y). Note that the skew value does not affect the timing of the trailing edge of the output pulse, but only the leading edge. Consequently the output pulse width is reduced by the skew value from Y to Y-X'. A maskable interrupt signal EVENT1 is generated when there is a match between EVCNT1 and ECMP1A.

A clearing signal on the EVI2 pin clears EVCNT2 in the PA unit and restarts the sequence.

ECMP2B can be used to generate an interrupt signal, EVENT2, when EVCNT2 accumulates a desired number of clock pulses.

11.2.2 Register Functions in Mode 0

11.2.2.1 Counter 1 (EVCNT1)

This counter drives the PWM section of the event counter. It is cleared by a successful comparison between ECMP2A and EVCNT2. The source chosen by input selector INPUT1 is used as the clock input to both counters (EVCNT1 and EVCNT2). The width and time of the output pulse is determined by the value in this counter matching the values in ECMP1A and ECMP1B.

11.2.2.2 Compare Register 1A (ECMP1A); (Y)

This compare register holds the nominal value of the output pulse width. Note that the actual output pulse width value will be equal to the value contained in ECMP1A minus the skew value which is held in event compare register 1B (ECMP1B). An interrupt signal, EVENT1 is generated when there is a match between ECMP1A and EVCNT1.

11.2.2.3 Compare Register 1B (ECMP1B); (X')

This compare register holds a value which adjusts the phase skew between the external signal (on EVI2) and the leading edge of the desired output signal. If the value in this register is zero, then the actual pulse width depends only on the value Y in ECMP1A. If the value in this register is equal to

the value Y in ECMP1A, the event output (EVO) will always be low (when EVPOL = 0). Note that this skew value should be a smaller value than the minimum value of the pulse width Y. (If the value in this register is larger than the value Y in ECMP1A, it is possible to miss a pulse in one cycle and the following cycle will be reversed in polarity cycle, which is equivalent to exchanging X' and Y in Figure 11-2.).

11.2.2.4 Counter 2 (EVCNT2)

This counter is cleared by the external signal on the EVI2 pin. The source chosen by input selector INPUT1 is used as the clock input to both counters (EVCNT1 and EVCNT2). This counter is used to control the amount of phase shift between the external signal's active edge and the start of the PWM output signal, using ECMP2A. A successful compare with compare register 2B will cause an interrupt request 2 (EVENT2) to be generated.

11.2.2.5 Compare Register 2A (ECMP2A); (X)

This compare register holds the value of the phase shift between the external signal active edge (or level) and the PWM output signal. Note that this phase shift value is added to the skew value held in ECMP1B to determine the timing of the leading edge of the PWM output pulse. If the value in this register is zero, then the active edge of the external signal on EVI2 will reset EVCNT1 immediately and there will be no phase shift, only skew.

11.2.2.6 Compare Register 2B (ECMP2B)

This compare register can be used to generate an interrupt when the PA unit has accumulated a desired number of clock pulses. When a match occurs between EVCNT2 and ECMP2B, an interrupt signal EVENT2 is generated which, if enabled, will interrupt the CPU.

11.2.2.7 Input Unit 1 (EVI1)

This unit selects the clock source for EVCNT1 and EVCNT2. If the input signal on the EVI1 pin (PH5) is used as a clock source, this unit will select the rising edge, falling edge or both edges of the input signal. If the E-clock or a scaled E-clock is selected as the clock source, this unit controls the active gated input level which inhibits counting. Even if PH5 is used for general purpose I/O, this unit is able to select the pass mode (EVI1C = 0, EVI1B = 0, EVI1A = 1).

11.2.2.8 Input Unit 2 (EVI2)

This unit selects the active edge or level of the input signal on the EVI2 pin (PH4) which will reset EVCNT2 (to zero).

11.2.2.9 Output Unit (EVO)

This unit controls the polarity of the PWM output signal. With the EVOEN (Event Output Enable) bit in the EVCTL register set to one, the EVPOL (Event Output Polarity) bit determines the polarity of the PWM output.

11.3 MODE 1: 8-BIT PWM AND PULSE ACCUMULATOR

In mode 1, the event counter operates as two independent 8-bit pulse accumulators or as one 8-bit pulse accumulator and one 8-bit PWM. The two counters work independently of each other. If mode 1 is selected (EVMDB = 0, EVMDA = 1), then the elements of the event counter are configured as shown in Figure 11-3.

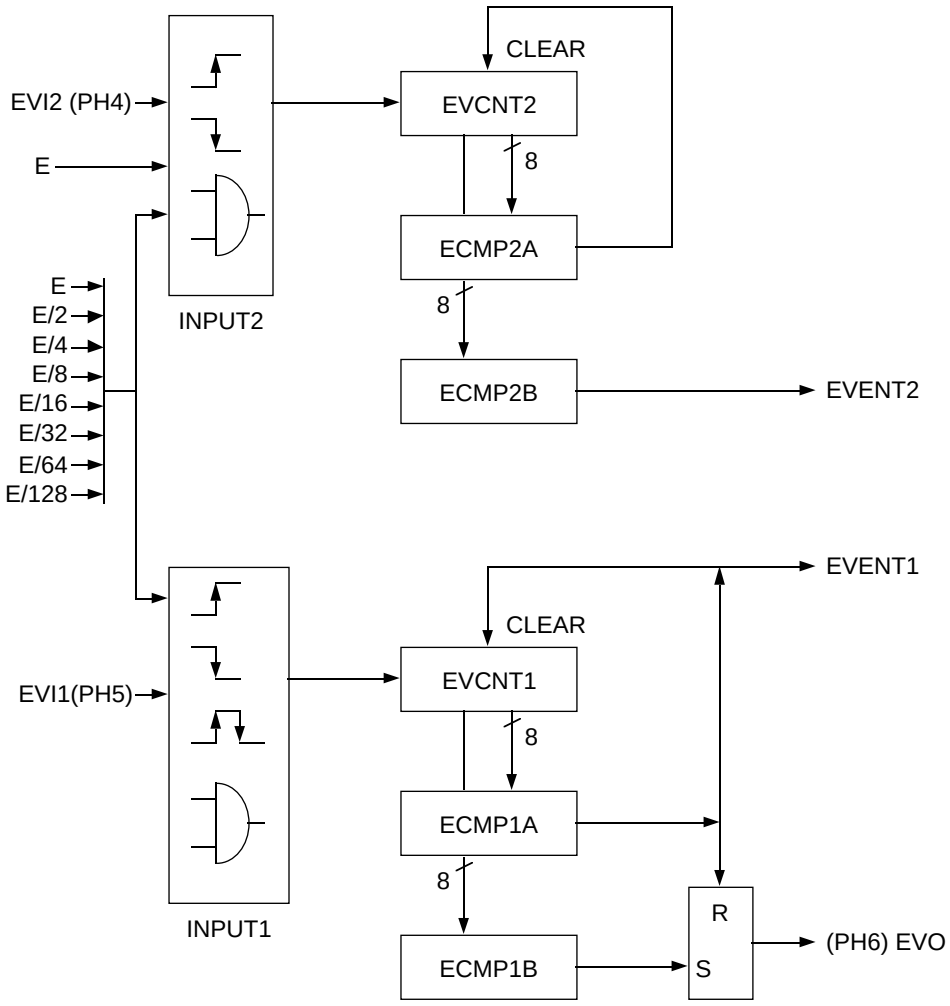


Figure 11-3. 8-bit PWM and Pulse Accumulator (Mode 1)

Both units can be driven by the E-clock, a scaled E-clock or by external signals applied to the EVI1 and EVI2 pins. However, note that only one scaled value of the E-clock can be selected. For example, if E/4 is selected as the clock source for EVCNT1, the only E-clock derivatives that can be chosen as the clock source for EVCNT2 are E and E/4. Alternatively, if E/4 is selected to drive EVCNT2, then E/4 is the only E-clock derivative that can be selected to drive EVCNT1.

11.3.1 Operation of Pulse Width Modulation Unit in Mode 1

The PWM unit can be clocked by the E-clock, a scaled E-clock, or an external signal applied to the EVI1 (PH5) pin. Alternatively an external signal on EVI1 can be used to gate the E-clock or scaled E-clock signal to the counter EVCNT1. The clock source is selected via the INPUT1 selector.

The period of the PWM output signal is determined by the value stored in ECMP1A. When a match occurs between this compare register and the counter register, EVCNT1, the output unit EVO is reset and EVCNT1 is cleared (to zero). At the same time, an interrupt signal EVENT1 is generated which, if enabled, interrupts the CPU.

The duty cycle of the PWM output signal is determined by the relationship between the values stored in ECMP1B and ECMP1A. The value stored in ECMP1B is the number of clock pulses to be accumulated by EVCNT1 before the output signal changes state. The output signal then remains in the new state for the remainder of the period, i.e. until EVCNT1 reaches the value stored in ECMP1A. The duty cycle can be expressed as a percentage of the period by the following equation:

$$\text{DUTY CYCLE} = 100 \times [1 - (\text{ECMP1B})/(\text{ECMP1A})] \%$$

The PWM unit will also work as a pulse accumulator by switching off the output signal to the EVO pin (PH6) and ignoring ECMP1B.

11.3.2 Operation of Pulse Accumulator Unit in Mode 1

The PA unit can be clocked by the E-clock, a scaled E-clock, or an external signal applied to the EVI2 (PH4) pin. Alternatively an external signal on EVI2 can be used to gate the E-clock or scaled E-clock signal to the counter EVCNT1. The clock source is selected via the INPUT2 selector.

EVCNT2 counts the pulses coming from the INPUT2 selector. It is cleared by a successful comparison between itself and the compare register ECMP2A. ECMP2A can be programmed with any 8-bit value to control the time when the counter is cleared.

ECMP2B can be used to generate an interrupt signal EVENT2 after a required number of input pulses have been accumulated by EVCNT2. If enabled, EVENT2 will interrupt the CPU.

11.3.3 Register Functions in Mode 1

11.3.3.1 Counter 1 (EVCNT1)

EVCNT1 counts the number of pulses coming from the INPUT1 selector. It is cleared by a successful comparison between itself and ECMP1A.

11.3.3.2 Compare Register 1A (ECMP1A)

This compare register holds the value of the PWM period which determines when interrupt 1 (EVENT1) occurs. On a successful comparison with EVCNT1, it generates EVENT1, clears EVCNT1 and resets output EVO to zero (when EVPOL = 0).

11.3.3.3 Compare Register 1B (ECMP1B)

This compare register holds the duty value which determines when the output (EVO) is set to one (when EVPOL = 0). The duty cycle can be expressed as a percentage of the period by the following equation:

$$\text{DUTY CYCLE} = 100 \times [1 - (\text{ECMP1B})/(\text{ECMP1A})]\ \%$$

11.3.3.4 Counter 2 (EVCNT2)

EVCNT2 counts the number of pulses coming from the INPUT2 selector. It is cleared by a successful comparison between itself and ECMP2A.

11.3.3.5 Compare Register 2A (ECMP2A)

This compare register holds the number of the pulses to be accumulated by EVCNT2 before EVCNT2 is cleared to zero.

11.3.3.6 Compare Register 2B (ECMP2B)

This compare register holds the number of the pulses to be accumulated by EVCNT2 before an interrupt signal EVENT2 is generated. If enabled, EVENT2 will interrupt the CPU.

11.3.3.7 Input Unit 1 (EVI1)

This input unit selects the clock source for EVCNT1. If the input signal EVI1 (PH5) is used as a clock source, this unit will select the rising edge, falling edge or both edges of the input signal. If the time-base counter is used as a clock source, this unit will select the active gated input level which inhibits counting while this input signal is at the active level.

11.3.3.8 Input Unit 2 (EVI2)

This input unit selects the clock source for EVCNT2. If the input signal EVI2 (PH4) is used as a clock source, this unit will select the rising edge or falling edge of the input signal. If the time base counter is used as a clock source, this unit will select the active gate input level which inhibits counting while this input signal is at the active level. Even if the time-base counter has selected one of the eight clock rates (E, E/2, E/4, E/8, E/16, E/32, E/64, or E/128), this input selector can select the E-clock independently as its clock source.

11.3.3.9 Output Unit (EVO)

This unit controls the output signal from the PWM unit. If the EVOEN bit in the EVCTL register is set to one, the EVO unit is enabled. The state of the EVPOL bit in the EVCTL register then determines the polarity of the event output. If EVO is not required for pulse width modulation, the EVOEN bit should be reset to zero to disable the EVO unit and to allow pin PH6 to be used for general purpose I/O.

11.4 MODE 2: 8-BIT PWM WITH PERIOD COUNTER

In mode 2, the event counter operates as an 8-bit PWM unit and an 8-bit counter (the PA unit) which increments at the end of each period of the PWM unit. If mode 2 is selected (EVMDB = 1, EVMDA = 0), then the elements of the event counter are configured as shown in Figure 11-4.

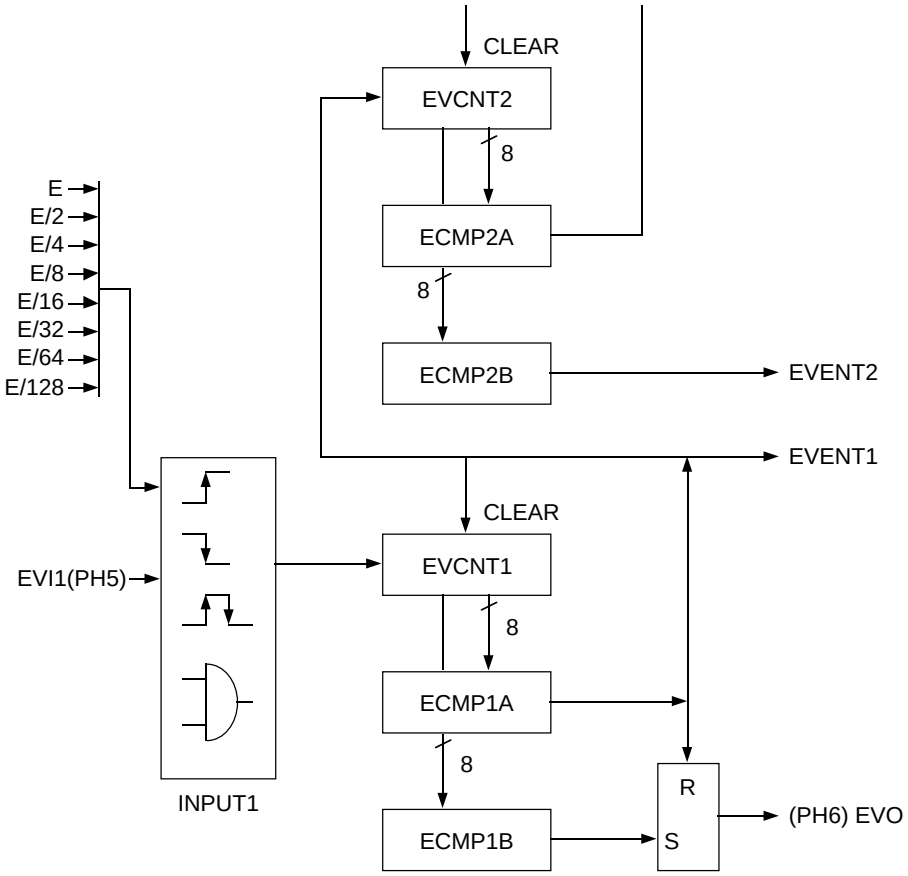


Figure 11-4. 8-bit PWM with Period Counter (Mode 2)

The only difference between this mode and mode 1 lies in the clock source used to drive the counter in the PA unit (EVCNT2). In mode 1, the clock signal comes via the INPUT1 selector and is either derived from the E-clock (gated or ungated) or an external signal. In mode 2, the signal which results from a successful comparison between EVCNT1 and ECMP1A in the PWM unit is used to drive EVCNT2 in the PA unit. This is the same signal which clears EVCNT1, thereby ending the PWM period. EVCNT2, therefore, counts the number of periods completed by the PWM unit.

11.4.1 Operation of Pulse Width Modulation Unit in Mode 2

The PWM unit can be clocked by the E-clock, a scaled E-clock, or an external signal applied to the EVI1 (PH5) pin. Alternatively an external signal on EVI1 can be used to gate the E-clock or scaled E-clock signal to the counter EVCNT1. The clock source is selected via the INPUT1 selector.

The period of the PWM output signal is stored in ECMP1A. When a match occurs between this compare register and the counter register, EVCNT1, the output unit EVO is reset, EVCNT1 is cleared (to zero) and EVCNT2 in the PA is incremented by one. At the same time, an interrupt signal EVENT1 is generated which, if enabled, interrupts the CPU.

The duty cycle of the PWM output signal is stored in ECMP1B. When a match occurs between this compare register and EVCNT1, the PWM output signal changes state (from zero to one or from one to zero depending on the polarity selected by the output unit.

$$\text{DUTY CYCLE} = 100 \times [1 - (\text{ECMP1B})/(\text{ECMP1A})]\ \%$$

The PWM unit will also work as a pulse accumulator by switching off the output signal to the EVO pin (PH6) and ignoring ECMP1B.

11.4.2 Operation of Pulse Accumulator Unit in Mode 2

The PA unit is clocked each time there is a successful comparison between EVCNT1 and ECMP1A in the PWM unit, i.e. at the end of each PWM period.

EVCNT2 counts the pulses coming from the PWM unit. It is cleared by a successful comparison between itself and the compare register ECMP2A. ECMP2A can be programmed with any 8-bit value to control the time when the counter is cleared.

ECMP2B can be used to generate an interrupt signal EVENT2 after a required number of input pulses have been accumulated by EVCNT2. If enabled, EVENT2 will interrupt the CPU.

11.4.3 Register Functions in Mode 2

11.4.3.1 Counter 1 (EVCNT1)

EVCNT1 counts the number of pulses coming from the INPUT1 selector. It is cleared by a successful comparison between itself and ECMP1A.

11.4.3.2 Compare Register 1A (ECMP1A)

This compare register holds the value of the PWM period which determines when interrupt 1 (EVENT1) occurs. On a successful comparison with EVCNT1, it generates EVENT1, clears EVCNT1, resets output EVO to zero (when EVPOL = 0) and clocks EVCNT2 in the PA unit.

11.4.3.3 Compare Register 1B (ECMP1B)

This compare register holds the PWM duty value which determines when the output (EVO) is set to one (when EVPOL = 0).

$$\text{DUTY CYCLE} = 100 \times [1 - (\text{ECMP1B})/(\text{ECMP1A})] \%$$

11.4.3.4 Counter 2 (EVCNT2)

EVCNT2 counts the number of pulses coming from the PWM unit. It is cleared when a successful comparison occurs between itself and ECMP2A.

11.4.3.5 Compare Register 2A (ECMP2A)

This compare register holds the number of the pulses to be accumulated by EVCNT2 before EVCNT2 is cleared to zero.

11.4.3.6 Compare Register 2B (ECMP2B)

This compare register holds the number of the pulses to be accumulated by EVCNT2 before an interrupt signal EVENT2 is generated. If enabled, EVENT2 will interrupt the CPU.

11.4.3.7 Input Unit 1 (EVI1)

This input unit selects the clock source for EVCNT1. If the input signal EVI1 (PH5) is used as a clock source, this unit will select the rising edge, falling edge or both edges of the input signal. If the time base counter is used as a clock source, this unit will select the active gated input level which inhibits counting while this input signal is at the active level.

11.4.3.8 Input Unit 2 (EVI2)

This unit is not used and the EVI2 pin (PH4) can be used as a general purpose I/O pin.

11.4.3.9 Output Unit (EVO)

This unit controls the output signal from the PWM unit. If the EVOEN bit in the EVCTL register is set to one, the EVO unit is enabled. The state of the EVPOL bit in the EVCTL register then determines the polarity of the event output. If EVO is not required for PWM, the EVOEN bit should be reset to zero to disable the EVO unit and to allow pin PH6 to be used for general purpose I/O.

11.5 MODE 3: 8-BIT PWM WITH 256 CLOCK PRESCALER

In mode 3, the event counter operates as an 8-bit PWM unit with a software programmable 8-bit clock prescaler (the PA unit). If mode 3 (EVMDB = 1, EVMDA = 1) is selected then the elements of the event counter are configured as shown in Figure 11-5.

The only difference between this mode and mode 1 lies in the clock source used to drive the counter in the PWM unit (EVCNT1). In mode 1, the clock signal comes via the INPUT2 selector and is either derived from the E-clock (gated or ungated) or an external signal. In mode 3, the signal which results from a successful comparison between EVCNT2 and ECMP2A in the PA unit is used to drive EVCNT1 in the PWM unit. EVCNT1, therefore, increments by one every time the value in EVCNT2 reaches the value programmed into ECMP2A. In other words, the PA unit acts as a prescaler to the PWM unit, its division ratio being the 8-bit value in ECMP2A.

11.5.1 Operation of Pulse Width Modulation Unit in Mode 3

The PWM unit is clocked each time there is a successful comparison between EVCNT2 and ECMP2A in the PA unit, i.e. at the end of each PA counting period.

The period of the PWM output signal is stored in ECMP1A. When a match occurs between this compare register and the counter register, EVCNT1, the output unit EVO is reset and EVCNT1 is cleared (to zero). At the same time, an interrupt signal EVENT1 is generated which, if enabled, interrupts the CPU.

The duty cycle of the PWM output signal is stored in ECMP1B. When a match occurs between this compare register and EVCNT1, the PWM output signal changes state (from zero to one or from one to zero depending on the polarity selected by the output unit.

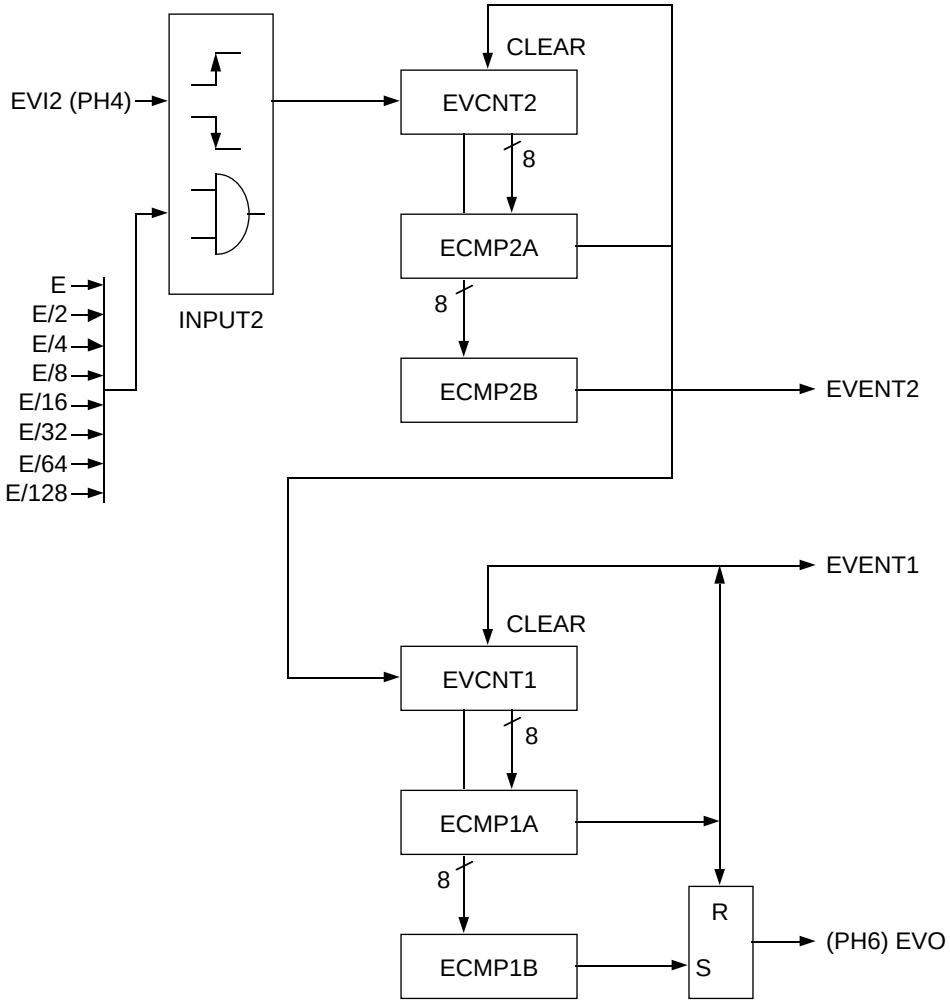


Figure 11-5. 8-bit PWM with 256 Clock Prescaler (Mode 3)

The PWM unit will also work as a pulse accumulator by switching off the output signal to the EVO pin (PH6) and ignoring ECMP1B.

11.5.2 Operation of Pulse Accumulator Unit in Mode 3

The PA unit can be clocked by the E-clock, a scaled E-clock, or an external signal applied to the EVI2 (PH4) pin. Alternatively an external signal on EVI2 can be used to gate the E-clock or scaled E-clock signal to the counter EVCNT2. The clock source is selected via the INPUT2 selector.

EVCNT2 counts the clock pulses coming from the INPUT2 selector. It is cleared by a successful comparison between itself and the compare register ECMP2A. ECMP2A can be programmed with

any 8-bit value to define when the counter is cleared and the PWM clocked. It causes the PA unit to act as a prescaler for the PWM by dividing the clock signal, coming via INPUT2, by any value between 1 and 256.

ECMP2B can be used to generate an interrupt signal EVENT2 after a required number of input pulses have been accumulated by EVCNT2. If enabled, EVENT2 will interrupt the CPU.

11.5.3 Register Functions in Mode 3

11.5.3.1 Counter 1 (EVCNT1)

EVCNT1 counts the number of pulses coming from the PA unit. It is cleared by a successful comparison between itself and ECMP1A.

11.5.3.2 Compare Register 1A (ECMP1A)

This compare register holds the value of the PWM period which determines when interrupt 1 (EVENT1) occurs. On a successful comparison with EVCNT1, it generates EVENT1, clears EVCNT1 and resets output EVO to zero (when EVPOL = 0).

11.5.3.3 Compare Register 1B (ECMP1B)

This compare register holds the PWM duty value which determines when the output (EVO) is set to one (when EVPOL = 0).

$$\text{DUTY CYCLE} = 100 \times [1 - (\text{ECMP1B})/(\text{ECMP1A})]\ \%$$

11.5.3.4 Counter 2 (EVCNT2)

EVCNT2 counts the number of clock pulses coming from the INPUT2 selector. It is cleared when a successful comparison occurs between itself and ECMP2A.

11.5.3.5 Compare Register 2A (ECMP2A)

This compare register holds the number of the pulses to be accumulated by EVCNT2 before EVCNT2 is cleared to zero.

11.5.3.6 Compare Register 2B (ECMP2B)

This compare register holds the number of the pulses to be accumulated by EVCNT2 before an interrupt signal EVENT2 is generated. If enabled, EVENT2 will interrupt the CPU.

11.5.3.7 Input Unit 1 (EVI1)

This unit is not used and the EVI1 pin (PH5) can be used as a general purpose I/O pin.

11.5.3.8 Input Unit 2 (EVI2)

This input unit selects the clock source for EVCNT2. If the input signal EVI2 (PH4) is used as a clock source, this unit will select the rising edge or falling edge of the input signal. If the time base counter is used as a clock source, this unit will select the active gate input level which inhibits counting while this input signal is at the active level. Even if the time base counter has selected one of the eight clock rates (E, E/2, E/4, E/8, E/16, E/32, E/64, or E/128), this input selector can select the E-clock independently as its clock source.

11.5.3.9 Output Unit (EVO)

This unit controls the output signal from the PWM unit. If the EVOEN bit in the EVCTL register is set to one, the EVO unit is enabled. The state of the EVPOL bit in the EVCTL register then determines the polarity of the event output. If EVO is not required for PWM, the EVOEN bit should be reset to zero to disable the EVO unit and to allow pin PH6 to be used for general purpose I/O.

11.6 EVENT COUNTER REGISTERS

11.6.1 Counter Clock Register (EVCLK)

	7	6	5	4	3	2	1	0	
$1070			EVMDB	EVMDA		EVCKC	EVCKB	EVCKA	EVCLK
RESET:	0	0	0	0	0	0	0	0	

READ: Any time.

WRITE: Any time.

RESET: $00.

EVMDB, EVMDA — Event Counter Mode bits

These two bits determine which event counter mode will be used.

EVMDB	EVMDA	MODE	
0	0	0	(8-bit PWM with selectable phase shift)
0	1	1	(8-bit PWM and 8-bit pulse accumulator)
1	0	2	(8-bit PWM with period counter)
1	1	3	(8-bit PWM with 256 clock prescaler)

EVCKC, EVCKB, EVCKA — Event Counter Prescaler bits

These three bits control the time-base counter. They select the prescaler value used to scale the E-clock before driving the event counters.

EVCKC	EVCKB	EVCKA	Prescaled Clock
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

11.6.2 Counter Control Register (EVCTL)

	7	6	5	4	3	2	1	0	
$1071	EVOEN	EVPOL	EVI2C	EVI2B	EVI2A	EVI1C	EVI1B	EVI1A	EVCTL
RESET:	0	0	0	0	0	0	0	0	

READ: Any time.

WRITE: Any time.

RESET: $00.

EVOEN — Event Output Enable (Port H, bit 6)

1 – PH6 is used as Event output.

0 – PH6 is used as a general purpose I/O.

EVPOL — Event Output Polarity

1 – Event Output (EVO) will be set (to one) when ECMP1A matches, and cleared (to zero) when ECMP1B matches.

0 – Event Output (EVO) will be set (to one) when ECMP1B matches, and cleared (to zero) when ECMP1A matches.

EVI2C, EVI2B, EVI2A — Event Input Select 2 (EVI2)

These three bits determine the function of event Input unit 2 as shown in the following tables.

The following table applies to modes 1 and 3 only:

EVI2C	EVI2B	EVI2A	PH4	Clock Source	Count Mode
0	0	0	I/O		Count stop
0	0	1	I/O	Scaled E	Count always
0	1	0	EVI2	Scaled E	Inhibit counting on EVI2 = 0
0	1	1	EVI2	Scaled E	Inhibit counting on EVI2 = 1
1	0	0	EVI2	External	Count on falling edge of EVI2
1	0	1	EVI2	External	Count on rising edge of EVI2
1	1	0	EVI2	E-Clock	Inhibit counting on EVI2 = 0
1	1	1	EVI2	E-Clock	Inhibit counting on EVI2 = 1

The following table applies to mode 0 only:

EVI2C	EVI2B	EVI2A	Clear input mode for EVCNT2
0	X	X	Not used
1	0	0	Falling edge of EVI2 (PH4)
1	0	1	Rising edge of EVI2 (PH4)
1	1	0	Logic HIGH (1) level on EVI2
1	1	1	Logic LOW (0) level on EVI2

In mode 0, EVI2 is used as a “clear” input to (EVCNT2).

Note: In mode 2, these three bits should be reset to zero to allow PH4 to be used as an I/O port.

EVI1C, EVI1B, EVI1A — Event Input Select 1 (EVI1)

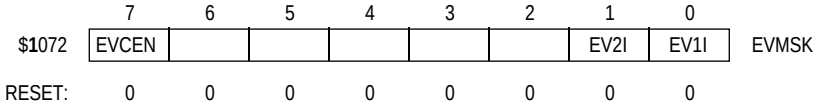
These three bits determine the operation of event input unit 1 as shown in the following table.

The following table applies to modes 0, 1 and 2 only:

EVI1C	EVI1B	EVI1A	PH5	Clock Source	Count Mode
0	0	0	I/O		Count stop
0	0	1	I/O	Scaled E	Count always
0	1	0	EVI1	Scaled E	Inhibit counting on EVI1 = 0
0	1	1	EVI1	Scaled E	Inhibit counting on EVI1 = 1
1	0	0	EVI1	External	Count on falling edge of EVI1
1	0	1	EVI1	External	Count on rising edge of EVI1
1	1	X	EVI1	External	Count on both falling and rising edges of EVI1

Note: In mode 3, these three bits should be reset to zero to allow PH5 to be used as an I/O port.

11.6.3 Counters Enable/Interrupt Mask Register (EVMSK)



READ: Any time.

WRITE: Any time.

RESET: $00.

EVCEN — Event Counters Enable

0 – Event counters EVCNT1 and EVCNT2 are cleared. Also the event output (EVO) is cleared to a logic low level (when EVPOL = 0 and EVOEN = 1). The EVCLK and EVCTL registers should be written while this bit is “0”.

1 – Event counters are enabled.

EV2I — Event 2 Interrupt Enable

0 – Interrupt inhibited.

1 – Hardware interrupt requested when EV2F flag set.

EV1I — Event 1 Interrupt Enable

0 – Interrupt inhibited.

1 – Hardware interrupt requested when EV1F flag set.

11.6.4 Interrupt Flag Register (EVFLG)

	7	6	5	4	3	2	1	0	
$1073							EV2F	EV1F	EVFLG
RESET:	0	0	0	0	0	0	0	0	

READ: Any time.

WRITE: Used as clearing mechanism (bits set cause corresponding bits to be cleared).

RESET: $00.

EV2F — Event Interrupt 2 Flag

Set by a successful match of EVCNT2 and ECMP2B. This bit is cleared automatically by a write to the EVFLG register with bit 1 set.

EV1F — Event interrupt 1 Flag

Set by a successful match of EVCNT1 and ECMP1A. This bit is cleared automatically by a write to the EVFLG register with bit 0 set.

11.6.5 Counter Registers (EVCNTx)

	7	6	5	4	3	2	1	0	
$1074	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0	EVCNT1
$1075	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0	EVCNT2
RESET:	U	U	U	U	U	U	U	U	

READ: Any time.

WRITE: Forces value to $00.

RESET: Indeterminate.

11.6.6 Counter Compare Registers (ECMPx)

	7	6	5	4	3	2	1	0	
$1076	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0	ECMP1A
$1077	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0	ECMP2A
$1078	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0	ECMP1B
$1079	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0	ECMP2B
RESET:	1	1	1	1	1	1	1	1	

READ: Any time.

WRITE: Any time.

RESET: $FF.

Compare registers ECMP1A and ECMP1B are associated with counter EVCNT1. Compare registers ECMP2A and ECMP2B are associated with counter EVCNT2.

11.7 PWM USING THE EVENT COUNTER

The PWM function of the event counter differs from the main PWM (discussed in Section 10) in that it has no double buffered duty and period registers. This means that it is possible to generate a cycle in which there is no duty change of state. This occurs when the duty value written is lower than the previous value, but the counter is already past the new value. In this case the counter register will increment all the way to the period count, roll over to $00 and count up to the new duty value before matching. This situation can be avoided by using the event interrupts to help calculate, in software, the correct values of duty and period to be written to the compare registers.

11.8 EFFECTIVE RANGE OF THE SET UP VALUES

In mode 0:

ECMP1A and ECMP1B: 0 to 255 ($00 to $FF)
ECMP2A and ECMP2B: 0 to 255 ($00 to $FF)

In modes 1, 2 and 3:

ECMP1A and ECMP1B: 1 to 255 ($01 to $FF)
(These values follow the boundary conditions of the PWM function.)

ECMP2A and ECMP2B: 1 to 255 to 256 ($01 to $FF to $00).

11.9 BOUNDARY CONDITIONS OF THE PWM FUNCTION

The PWM function (In modes 1, 2, and 3) follows the boundary conditions described below. The period is determined by the value in the ECMP1A register (8-bit). The duty value is determined by the values in the ECMP1A and ECMP1B registers (8-bit).

If duty = 0, period > 0, EVPOL = 0 and EVOEN = 1;	PH6 is always high.
If duty = 0, period > 0, EVPOL = 1 and EVOEN = 1;	PH6 is always low.
If duty ≥ period, EVPOL = 0 and EVOEN =1;	PH6 is always low.
If duty ≥ period, EVPOL = 1 and EVOEN =1;	PH6 is always high.
If period = 0, EVPOL = 0 and EVOEN =1;	PH6 is always low.
If period = 0, EVPOL = 1 and EVOEN =1;	PH6 is always high.
If EVOEN = 0;	PH6 is I/O port under the control of DDRH bit 6.

SECTION 12

CPU, ADDRESSING MODES AND INSTRUCTION SET

This section discusses the M68HC11 central processing unit (CPU) architecture, its addressing modes and the instruction set (by instruction type). Everything discussed in this section applies to the MC68HC11G5. For more detailed information on the instruction set, refer to the M68HC11 Reference Manual (M68HC11RM/D).

12.1 PROGRAMMING MODEL AND CPU REGISTERS

In addition to being able to execute all M6800 and M6801 instructions, the MC68HC11G5 uses a 4-page opcode map to allow execution of 91 new opcodes (see **12.3: INSTRUCTION SET**). Seven registers, shown in Figure 12-1 and discussed in the following paragraphs, are available to programmers.

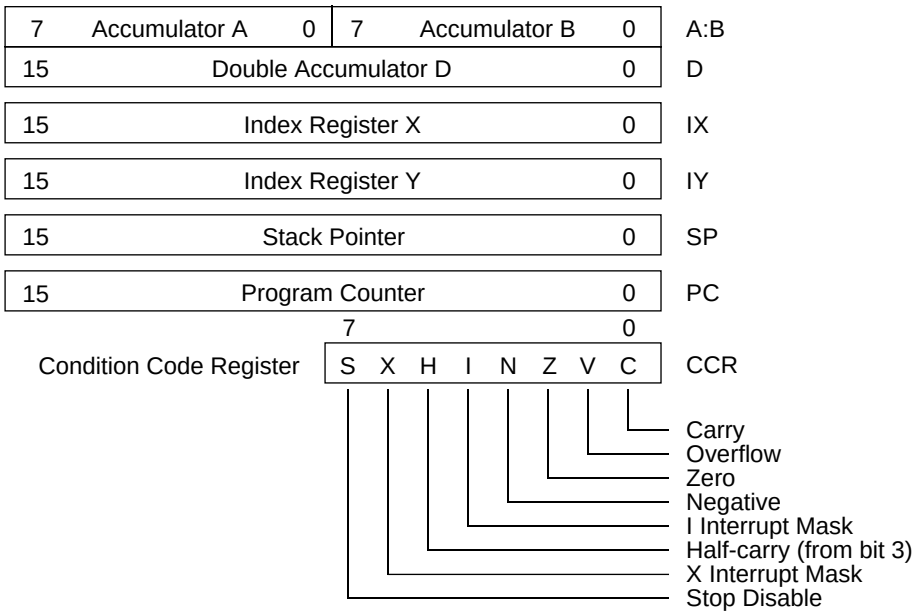


Figure 12-1. Programming Model

12.1.1 Accumulators A and B

Accumulator A and accumulator B are general purpose 8-bit registers used to hold operands and results of arithmetic calculations or data manipulations. These two accumulators can be concatenated into a single 16-bit accumulator called accumulator D.

12.1.2 Index Register X (IX)

The 16-bit IX register is used for indexed mode addressing. It provides a 16-bit indexing value which is added to an 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.

12.1.3 Index Register Y (IY)

The 16-bit IY register is also used for indexed mode addressing, similar to the IX register; however, all instructions using the IY register have a 2-byte opcode and require an extra cycle of execution time.

12.1.4 Stack Pointer (SP)

The stack pointer (SP) is a 16-bit register which contains the address of the next free location on the stack. The stack is configured as a sequence of last-in/first-out read/write registers which allow important data to be stored during interrupts and subroutine calls. Each time a new byte is added to the stack (a push), the SP is decremented; each time a byte is removed from the stack (a pull) the SP is incremented.

12.1.5 Program Counter (PC)

The program counter is a 16-bit register which contains the address of the next instruction to be executed.

12.1.6 Condition Code Register (CCR)

The condition code register is an 8-bit register in which the bits signify the results of the instruction just executed. These bits can be individually tested by software and a specific action can be taken as a result of the test. Each individual condition code register bit is explained below.

Carry/Borrow (C) — The C bit is set if there was a carry or borrow out of the arithmetic logic unit (ALU) during the last arithmetic operation. The C bit is also affected during shift and rotate instructions.

Overflow (V) — The overflow bit is set if there was an arithmetic overflow as a result of the operation; otherwise, the V bit is cleared.

Zero (Z) — The zero bit is set if the result of the last arithmetic, logic or data manipulation operation was zero; otherwise, the Z bit is cleared.

Negative (N) — The negative bit is set if the result of the last arithmetic, logic or data manipulation operation was negative; otherwise, the N bit is cleared. A result is said to be negative if its most significant bit is set to one.

I Interrupt Mask (I) — The I interrupt mask bit is set, either by hardware or software, to disable (mask) all maskable interrupt sources (both external and internal).

Half Carry (H) — The half carry bit is set when a carry occurs between bits 3 and 4 of the ALU during an ADD, ABA or ADC instruction; otherwise, the H bit is cleared.

X Interrupt Mask (X) — The X interrupt mask bit is set only by hardware ($\overline{\text{RESET}}$ or $\overline{\text{XIRQ}}$ acknowledge) and is cleared only by a software instruction (TAP or RTI).

Stop Disable (S) — The stop disable bit is set to disable the STOP instruction, and cleared to enable the STOP instruction. The S bit is program controlled. The STOP instruction is treated as no operation (NOP) if the S bit is set.

12.2 ADDRESSING MODES

In the M68HC11 CPU, six addressing modes can be used to reference memory; immediate, direct, extended, indexed (with either of two 16-bit index registers and an 8-bit offset), inherent, and relative. Some instructions require an additional byte before the opcode to accommodate a multi-page opcode map; this byte is called a pre-byte.

The following paragraphs provide a description of each addressing mode. In these descriptions the term “effective address” is used to indicate the memory address from which the argument is fetched or stored, or from which execution is to proceed.

12.2.1 Immediate Addressing (IMM)

In the immediate addressing mode, the actual argument is contained in the byte(s) immediately following the instruction, where the number of bytes matches the size of the register. These are two, three, or four (if pre-byte is required) byte instructions.

12.2.2 Direct Addressing (DIR)

In the direct addressing mode (sometimes called page zero addressing), the least significant byte of the operand address is contained in a single byte following the opcode. The high order byte of the effective address is assumed to be $00 and is not included as an instruction byte. Direct addressing allows the user to access $0000 through $00FF using 2-byte instructions and execution time is reduced by eliminating the additional memory access. In most applications, this 256-byte area is reserved for frequently referenced data. In the MC68HC11G5, software can configure the memory map so that internal RAM, and/or internal registers or external memory can occupy these addresses.

12.2.3 Extended Addressing (EXT)

In the extended addressing mode, the effective address of the instruction appears explicitly in the two bytes following the opcode. Therefore, the length of most instructions using the extended addressing mode is three bytes: one for the opcode and two for the effective address.

12.2.4 Indexed Addressing (IND, X; IND, Y)

In the indexed addressing mode, either the X or Y index register is used in calculating the effective address. In this case, the effective address is variable and depends on the current contents of the X or Y index register and a fixed 8-bit unsigned offset contained in the instruction. This addressing mode can be used to reference any memory location in the 64 kbyte address space. These are usually two (or three if a pre-byte is required) byte instructions, the opcode plus the 8-bit offset.

12.2.5 Inherent Addressing (INH)

In the inherent addressing mode, all of the information is contained in the opcode. The operands (if any) are registers and no memory reference is required. These are usually one or two byte instructions.

12.2.6 Relative Addressing (REL)

The relative addressing mode is used for branch instructions. If the branch condition is true, the contents of the 8-bit signed byte following the opcode (the offset) is added to the contents of the program counter to form the effective branch address; otherwise, control proceeds to the instruction immediately following the branch instruction. These are usually two byte instructions.

12.3 INSTRUCTION SET

This section explains the basic capabilities and organization of the instruction set. For this discussion the instruction set is divided into functional groups of instructions. Some instructions will appear in more than one functional group. For example, transfer accumulator A to condition code register (TAP) appears in the condition-code-register group and in the load/store/transfer subgroup of accumulator/memory instructions. For a detailed explanation of each instruction refer to the M68HC11 Reference Manual (M68HC11RM/D).

In order to expand the number of instructions used in the MC68HC11G5, a pre-byte mechanism has been added which affects certain instructions. Most of the instructions affected are associated with the Y index register. Instructions which do not require a pre-byte are said to reside in page 1 of the opcode map. Instructions requiring a pre-byte are said to reside in pages 2, 3, and 4 of the opcode map. The opcode map pre-byte codes are $18 for page 2, $1A for page 3, and $CD for page 4. A pre-byte code applies only to the opcode which immediately follows it. That is, all instructions are assumed to be single byte opcodes unless the first byte of the instruction happens to correspond to one of the three pre-byte codes rather than a page 1 opcode.

12.3.1 Accumulator and Memory Instructions

Most of these instructions use two operands. One operand is either an accumulator or an index register while the second operand is usually obtained from memory using one of the addressing modes discussed earlier. These accumulator memory instructions can be divided into six subgroups:

1. Loads, stores and transfers,
2. Arithmetic operations,
3. Multiply and divide,
4. Logical operations,
5. Data testing and bit manipulation,
6. Shifts and rotates.

These instructions are discussed in the following tables and paragraphs.

12.3.1.1 Loads, Stores and Transfers

Almost all MCU activities involve moving data from memories or peripherals into the CPU or moving results from the CPU to memory or to I/O devices. The load, store, and transfer instructions associated with the accumulators are summarized in the following table. There are additional load, store, push, and pull instructions associated with the index registers and stack pointer register (see **12.3.2 Stack And Index Register Instructions**).

Table 12-1. Loads, Stores and Transfers

Function	Mnemonic	IMM	DIR	EXT	INDX	INDY	INH
Clear Memory Byte	CLR			X	X	X	
Clear Accumulator A	CLRA						X
Clear Accumulator B	CLRB						X
Load Accumulator A	LDAA	X	X	X	X	X	
Load Accumulator B	LDAB	X	X	X	X	X	
Load Double Accumulator D	LDD	X	X	X	X	X	
Pull A from Stack	PULA						X
Pull B from Stack	PULB						X
Push A onto Stack	PSHA						X
Push B onto Stack	PSHB						X
Store Accumulator A	STAA	X	X	X	X	X	
Store Accumulator B	STAB	X	X	X	X	X	
Store Double Accumulator D	STD	X	X	X	X	X	
Transfer A to B	TAB						X
Transfer A to CCR	TAP						X
Transfer B to A	TBA						X
Transfer CCR to A	TPA						X
Exchange D with X	XGDX						X
Exchange D with Y	XGDY						X

12.3.1.2 Arithmetic Operations

This group of instructions supports arithmetic operations on a variety of operands. 8 and 16-bit operations are supported directly and can easily be extended to support multiple word operands. Twos complement (signed) and binary (unsigned) operations are supported directly. BCD arithmetic is supported by following normal arithmetic instruction sequences with the decimal adjust accumulator A (DAA) instruction to restore results to BCD format. Compare instructions perform a subtraction within the CPU to update the condition code bits without altering either operand. Test instructions are provided but are seldom needed since almost all other operations automatically update the condition code bits anyway.

Table 12-2. Arithmetic Operations

Function	Mnemonic	IMM	DIR	EXT	INDX	INDY	INH
Add Accumulators	ABA						X
Add Accumulator B to X	ABX						X
Add Accumulator B to Y	ABY						X
Add with Carry to A	ADCA	X	X	X	X	X	
Add with Carry to B	ADCB	X	X	X	X	X	
Add Memory to A	ADDA	X	X	X	X	X	
Add Memory to B	ADDB	X	X	X	X	X	
Add Memory to D (16-bit	ADDD	X	X	X	X	X	
Compare A to B	CBA						X
Compare A to Memory	CMPA	X	X	X	X	X	
Compare B to Memory	CMPB	X	X	X	X	X	
Compare D to Memory (16-bit	CPD	X	X	X	X	X	
Decimal Adjust A (for BCD	DAA						X
Decrement Memory Byte	DEC			X	X	X	
Decrement Accumulator A	DECA						X
Decrement Accumulator B	DECB						X
Increment Memory Byte	INC			X	X	X	
Increment Accumulator A	INCA						X
Increment Accumulator B	INCB						X
Twos Complement Memory Byte	NEG			X	X	X	
Twos Complement Accumulator A	NEGA						X
Twos Complement Accumulator B	NEGB						X
Subtract with Carry from A	SBCA	X	X	X	X	X	
Subtract with Carry from B	SBCB	X	X	X	X	X	
Subtract Memory from A	SUBA	X	X	X	X	X	
Subtract Memory From B	SUBB	X	X	X	X	X	
Subtract Memory From D (16-bit	SUBD	X	X	X	X	X	
Test Memory for Zero or Minus	TST			X	X	X	
Test A for Zero or Minus	TSTA						X
Test B for Zero or Minus	TSTB						X

12.3.1.3 Multiply and Divide

One multiply and two divide instructions are provided. The 8-bit by 8-bit multiply instruction (MUL) produces a 16-bit result. The integer divide (IDIV) performs a 16-bit by 16-bit divide and produces a 16-bit result and a 16-bit remainder. The fractional divide (FDIV) divides a 16-bit numerator by a larger 16-bit denominator to produce a 16-bit result (a binary weighted fraction between 0 and 0.99998) and a 16-bit remainder. FDIV can be used to further resolve the remainder from an IDIV or FDIV operation.

Table 12-3. Multiply and Divide

Function	Mnemonic	INH
Multiply (A x B ⇨ D)	MUL	X
Fractional Divide (D ÷) ⇨ X; r ⇨ D)	FDIV	X
Integer Divide (D ÷) ⇨ X; r ⇨ D)	IDIV	X

12.3.1.4 Logical Operations

This group of instructions is used to perform the boolean logical operations AND, inclusive-OR, exclusive-OR, and complement.

Table 12-4. Logical Operations

Function	Mnemonic	IMM	DIR	EXT	INDX	INDY	INH
AND A with Memory	ANDA	X	X	X	X	X	
AND B with Memory	ANDB	X	X	X	X	X	
Bit(s) Test A with Memory	BITA	X	X	X	X	X	
Bit(s) Test B with Memory	BITB	X	X	X	X	X	
Ones Complement Memory Byte	COM			X	X	X	
Ones Complement A	COMA						X
Ones Complement B	COMB						X
OR A with Memory (Exclusive	EORA	X	X	X	X	X	
OR B with Memory (Exclusive	EORB	X	X	X	X	X	
OR A with Memory (Inclusive	ORAA	X	X	X	X	X	
OR B with Memory (Inclusive	ORAB	X	X	X	X	X	

12.3.1.5 Data Testing and Bit Manipulation

This group of instructions is used to operate on operands as small as a single bit, but these instructions can also operate on any combination of bits within any 8-bit location in the 64 kbyte memory space. The bit test (BITA or BITB) instructions perform an AND operation within the CPU to update condition code bits without altering either operand. The BSET and BCLR instructions read the operand, manipulate selected bits within the operand, and write the result back to the operand address. Some care is required when read-modify-write instructions such as BSET and BCLR are used on I/O and control register locations because the physical location read is not always the same as the location written.

Table 12-5. Data Testing and Bit Manipulation

Function	Mnemonic	IMM	DIR	EXT	INDX	INDY
Bit(s) Test A with Memory	BITA	X	X	X	X	X
Bit(s) Test B with Memory	BITB	X	X	X	X	X
Clear Bit(s) in Memory	BCLR		X		X	X
Set Bit(s) in Memory	BSET		X		X	X
Branch if Bit(s) Clea	BRCLR		X		X	X
Branch if Bit(s) Se	BRSET		X		X	X

12.3.1.6 Shifts and Rotates

The shift and rotate functions in the M68HC11 CPU all involve the carry bit in the condition code register in addition to the 8 or 16-bit operand in the instruction. This permits easy extension to multiple word operands. Also, by setting or clearing the carry bit before a shift or rotate instruction, the programmer can easily control what will be shifted into the opened end of an operand. The ASR instruction maintains the original value of the most significant bit of the operand which facilitates manipulation of twos complement (signed) numbers.

Table 12-6. Shifts and Rotates

Function	Mnemonic	IMM	DIR	EXT	INDX	INDY	INH
Arithmetic Shift Left Memory	ASL			X	X	X	
Arithmetic Shift Left A	ASLA						X
Arithmetic Shift Left B	ASLB						X
Arithmetic Shift Left Double	ASLD						X
Arithmetic Shift Right Memory	ASR			X	X	X	
Arithmetic Shift Right A	ASRA						X
Arithmetic Shift Right B	ASRB						X
(Logical Shift Left Memory)	(LSL)			X	X	X	
(Logical Shift Left A	(LSLA)						X
(Logical Shift Left B	(LSLB)						X
(Logical Shift Left Double	(LSLD)						X
Logical Shift Right Memory	LSR			X	X	X	
Logical Shift Right A	LSRA						X
Logical Shift Right B	LSRB						X
Logical Shift Right D	LSRD						X
Rotate Left Memory	ROL			X	X	X	
Rotate Left A	ROLA						X
Rotate Left B	ROLB						X
Rotate Right Memory	ROR			X	X	X	
Rotate Right A	RORA						X
Rotate Right B	RORB						X

The logical left shift instructions are shown in parenthesis because there is no difference between an arithmetic and a logical left shift. Both mnemonics are recognized by the assembler as equivalent, but having both instruction mnemonics makes some programs easier to read.

12.3.2 Stack and Index Register Instructions

The following table summarizes the instructions available for the 16-bit index registers (X and Y) and the 16-bit stack pointer.

Table 12-7. Stack And Index Register Instructions

Function	Mnemonic	IMM	DIR	EXT	INDX	INDY	INH
Add Accumulator B to X	ABX						X
Add Accumulator B to Y	ABY						X
Compare X to Memory (16 Bit)	CPX	X	X	X	X	X	
Compare Y to Memory (16 Bit)	CPY	X	X	X	X	X	
Decrement Stack Pointer	DES						X
Decrement Index Register X	DEX						X
Decrement Index Register Y	DEY						X
Increment Stack Pointer	INS						X
Increment Index Register X	INX						X
Increment Index Register Y	INY						X
Load Index Register X	LDX	X	X	X	X	X	
Load Index Register Y	LDY	X	X	X	X	X	
Load Stack Pointer	LDS	X	X	X	X	X	
Pull X from Stack	PULX						X
Pull Y from Stack	PULY						X
Push X onto Stack	PSHX						X
Push Y onto Stack	PSHY						X
Store Index Register X	STX	X	X	X	X	X	
Store Index Register Y	STY	X	X	X	X	X	
Store Stack Pointer	STS	X	X	X	X	X	
Transfer SP to X	TSX						X
Transfer SP to Y	TSY						X
Transfer X to SP	TXS						X
Transfer Y to SP	TYS						X
Exchange D with X	XGDX						X
Exchange D with Y	XGDY						X

The exchange D with X (XGDX) and exchange D with Y (XGDY) instructions provide a simple way of getting a pointer value from a 16-bit index register to the D accumulator which has more powerful 16-bit arithmetic capabilities than the 16-bit index registers. Since these are bidirectional exchanges,

the original value of the D accumulator is automatically preserved in the index register while the pointer is being manipulated in the D accumulator. When pointer calculations are finished, another exchange simultaneously updates the index register and restores the D accumulator to its former value.

The transfers between an index register and the stack pointer deserve additional comment. The stack pointer always points at the next free location on the stack as opposed to the last thing that was pushed onto the stack. The usual reason for transferring the stack pointer value into an index register is to allow indexed addressing access to information that was formerly pushed onto the stack. In such cases, the address pointed-to by the stack pointer is of no value since nothing has been stored at that location yet. This explains why the value in the stack pointer is incremented during transfers to an index register. There is a corresponding decrement of a 16-bit value as it is transferred from an index register to the stack pointer.

12.3.3 Condition Code Register Instructions

This group of instructions allows the programmer to manipulate bits in the condition code register.

Table 12-8. Condition Code Register Instructions

Function	Mnemonic	INH
Clear Carry Bi	CLC	X
Clear Interrupt Mask Bi	CLI	X
Clear Overflow Bit	CLV	X
Set Carry Bi	SEC	X
Set Interrupt Mask Bit	SEI	X
Set Overflow Bit	SEV	X
Transfer A to CCF	TAP	X
Transfer CCR to A	TPA	X

At first look, it may appear that there should be a set and a clear instruction for each of the 8 bits in the condition code register while these instructions are present for only 3 of the 8 bits (C, I and V). Upon closer consideration there are good reasons for not including the set and clear instructions for the other five bits.

The stop disable (S) bit is an unusual case because this bit is intended to lock out the STOP instruction for those who view it as an undesirable function in their application. Providing set and clear instructions for this bit would make it easier to enable STOP when it was not wanted or disable STOP when it was wanted. The TAP instruction provides a way to change S but cuts the chances of an undesirable change to S in half because the value of the A accumulator at the time the TAP instruction is executed determines whether or not S will actually change.

The $\overline{\text{XIRQ}}$ interrupt mask (X) bit is another unusual case. The definition of this bit specifically states that software shall not be allowed to change X from 0 to 1, in fact, this is even prohibited by hardware logic. This immediately eliminates a need for a set X instruction. For arguments similar to those used for the S bit, the TAP instruction is preferred over a clear X instruction as a means of clearing X because TAP makes it a little less likely that X will become cleared before the programmer really intended to clear it.

The half-carry (H) bit needs no set or clear instructions because this condition code bit is only used by the DAA instruction to adjust the result of a BCD add or subtract. The H bit is not used as a test condition for any branches so it would not be useful to be able to set or clear this bit.

This leaves only the negative (N) and zero (Z) condition code bits. In contrast to S, X, and H, it is often useful to be able to easily set or clear these flag bits. A clear accumulator instruction such as CLRB will clear both the N and Z condition code bits. The instruction “LDAA #$80” causes both N and Z to be set. Since there are so many simple instructions that can accomplish setting or clearing of N and Z, it is not necessary to provide specific set and clear instructions for N and Z in this group.

12.3.4 Program Control Instructions

This group of instructions is used to control the flow of a program rather than to manipulate data. Since this group is so large it has been further divided into five subgroups:

1. Branches,
2. Jumps,
3. Subroutine calls and returns,
4. Interrupt handling,
5. Miscellaneous.

12.3.4.1 Branches

These instructions allow the CPU to make decisions based on the contents of the condition code bits. All decision blocks in a flow chart would correspond to one of the conditional branch instructions which are summarized in the following table.

Table 12-9. Branches

Functior	Mnemonic	REL	DIR	INDX	INDY	Comments
Branch if Carry Clea	BCC	X				C = 0 ?
Branch if Carry Set	BCS	X				C = 1 ?
Branch if Equal Zerc	BEQ	X				Z = 1 ?
Branch if Greater Than or Equa	BGE	X				Signed≥
Branch if Greater Than	BGT	X				Signed >
Branch if Higher	BHI	X				Unsigned >
Branch if Higher or Same (same as BCC)	BHS	X				Unsigned≥
Branch if Less Than or Equa	BLE	X				Signed ≤
Branch if Lower (same as BCS)	BLO	X				Unsigned <
Branch if Lower or Same	BLS	X				Unsigned ≤
Branch if Less Than	BLT	X				Signed <
Branch if Minus	BMI	X				N = 1 ?
Branch if Not Equal	BNE	X				Z = 0 ?
Branch if Plus	BPL	X				N = 0 ?
Branch if Bit(s) Clear in Memory Byte	BRCLR		X	X	X	Bit Manipulation
Branch Never	BRN	X				3-cycle NOP
Branch if Bit(s) Set in Memory Byte	BRSET		X	X	X	Bit Manipulation
Branch if Overflow Clear	BVC	X				V = 0 ?
Branch if Overflow Set	BVS	X				V = 1 ?

The limited range of branches (-128/+127 locations) is more than adequate for most (but not all) situations. In cases where this range is too short, a jump instruction must be used. For every branch there is a branch for the opposite condition so it is a simple matter to replace a branch which has an out-of-range destination with a sequence consisting of the opposite branch around a jump to the out-of-range destination. For example, if a program contained the instruction...

```
        BHI     TINBUK2   Unsigned >
```

where TINBUK2 was out of the -128/+127 location range, the following instruction sequence could be substituted...

```
        BLS     AROUND    Unsigned ≤
        JMP     TINBUK2   Still go to TINBUK2 if >
AROUND  EQU     *
```

12.3.4.2 Jumps

The jump instruction allows control to be passed to any address in the 64 kbyte memory map.

Table 12-10. Jumps

Function	Mnemonic	DIR	EXT	INDX	INDY	INH
Jump	JMP	X	X	X	X	

12.3.4.3 Subroutine Calls and Returns (BSR, JSR, RTS)

These instructions provide an easy way to break a programming task into manageable blocks called subroutines. The CPU automates the process of remembering the address in the main program where processing should resume after the subroutine is finished. This address is automatically pushed onto the stack when the subroutine is called and is pulled off the stack during the return from subroutine (RTS) instruction which ends the subroutine.

Table 12-11. Subroutine Calls and Returns (BSR, JSR, RTS)

Function	Mnemonic	REL	DIR	EXT	INDX	INDY	INH
Branch to Subroutine	BSR	X					
Jump to Subroutine	JSR		X	X	X	X	
Return from Subroutine	RTS						X

12.3.4.4 Interrupt Handling (RTI, SWI, WAI)

This group of instructions is related to interrupt operations.

Table 12-12. Interrupt Handling (RTI, SWI, WAI)

Function	Mnemonic	INH
Return from Interrupt	RTI	X
Software Interrupt	SWI	X
Wait for Interrupt	WAI	X

The software interrupt (SWI) instruction is similar to a jump to subroutine (JSR) instruction except that the contents of all working CPU registers are saved on the stack, rather than just saving the return address. SWI is unusual in that it is requested by the software program as opposed to other interrupts which are requested asynchronously with respect to the executing program.

Wait for interrupt (WAI) has two main purposes. WAI is executed to place the MCU in a reduced power consumption standby state (WAIT mode) until some interrupt occurs. The other use is to reduce the latency time to some important interrupt. The reduction of latency comes about because the time consuming task of storing the CPU registers on the stack is done as soon as the WAI instruction starts to execute. When the interrupt finally comes, the CPU is ready to fetch the appropriate vector so the delay associated with register stacking is eliminated from latency calculations.

12.3.4.5 Miscellaneous (NOP, STOP, TEST)

NOP can be used to introduce a small time delay into the flow of a program. This is often useful in meeting the timing requirements of slow peripherals. By incorporating NOP instructions into loops, longer delays can be produced.

Table 12-13. Miscellaneous (NOP, STOP, TEST)

Function	Mnemonic	INH
No Operation (2-cycle delay	NOP	X
Stop Clocks	STOP	X
Test	TEST	X

During debugging it is common to replace various instructions with NOP opcodes to effectively remove an unwanted instruction without having to rearrange the rest of the program.

Occasionally a programmer is faced with the problem of fine tuning the delays through various paths in his program. In such cases it is sometimes useful to use a BRN instruction as a 3 cycle NOP. It is also possible to fine tune execution time by choosing alternate addressing mode variations of instructions to change the execution time of an instruction sequence without changing the program's function.

STOP causes the oscillator and all MCU clocks to freeze. This frozen state is called STOP mode and power consumption is dramatically reduced in this mode. The operation of this instruction is also dependent upon the S condition code bit because the STOP mode is not appropriate for all applications. If S is 1 the STOP instruction is treated as a no operation (NOP) instruction and processing just continues to the next instruction.

The TEST instruction is used only during factory testing and is treated as an illegal opcode in normal operating modes of the MCU. This instruction causes unusual behaviour on the address bus (counts backwards) which prevents its use in any normal system.

This page intentionally left blank

SECTION 13

ELECTRICAL SPECIFICATIONS

This section contains the electrical specifications and associated timing information for the MC68HC11G5.

13.1 MAXIMUM RATINGS †

Rating	Symbol	Value	Unit
Supply Voltage	V_{DD}	– 0.5 to 7.0	V
Input Voltage	V_{in}	V_{SS} – 0.5 to V_{DD} + 0.5	V
Operating Temperature Range	T_A	T_L to T_H – 40 to 85	°C
Storage Temperature Range	T_{stg}	– 65 to 150	°C
Current Drain per Pin * Excluding V_{DD} and V_{SS}	I_D	25	mA

* One pin at a time, observing maximum power dissipation limits.

† This device contains protective circuitry against damage due to high static voltages or electrical fields; however, it is advised that normal precautions be taken to avoid application of any voltages higher than maximum-rated voltages to this high-impedance circuit. Reliability of operation is enhanced if unused inputs are tied to an appropriate logic voltage level (eg. either GND or V_{DD}).

13.2 THERMAL CHARACTERISTICS

Characteristic	Symbol	Value	Unit
Thermal Resistance Plastic	θ_{JA}	50	°C/W

13.3 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 (^oC)
θ_{JA} = Package thermal resistance, junction-to-ambient ($^oC/W$)
P_D = $P_{INT} + P_{I/O}$
P_{INT} = Internal chip power = $I_{DD} \times V_{DD}$ (W)
$P_{I/O}$ = Power dissipation on input and output pins (W) — user determined)

Note: For most applications $P_{I/O} < P_{INT}$ and can be neglected.

An approximate relationship between P_D and T_J (if $P_{I/O}$ is neglected) is:

$$P_D = K \div (T_J + 273^oC) \quad (2)$$

Solving equations (1) and (2) for K gives:

$$K = P_D \bullet (T_A + 273^oC) + \theta_{JA} \bullet P_D^2 \quad (3)$$

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

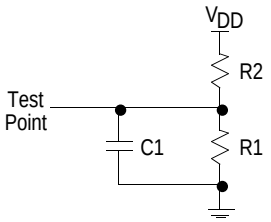
13.4 DC ELECTRICAL CHARACTERISTICS

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

Characteristic		Symbol	Min	Max	Unit
Output Voltage	All outputs	V_{OL}	—	0.1	V
I_{Load} = ± 10.0µA (See Note 1)	All outputs except $\overline{RESET}$ and MODA	V_{OH}	V_{DD} - 0.1	—	
Output High Voltage	All outputs except	V_{OH}	V_{DD} - 0.8	—	V
I_{Load} = - 0.8mA, V_{DD} = 4.5V (See Note 1)	$\overline{RESET}$, XTAL and MODA				
Output Low Voltage	All outputs except XTAL	V_{OL}	—	0.4	V
I_{Load} = 1.6mA					
Input High Voltage	$\overline{RESET}$	V_{IH}	0.8 x V_{DD}	V_{DD}	V
	All inputs except $\overline{RESET}$	V_{IH}	0.7 x V_{DD}	V_{DD}	
Input Low Voltage	All Inputs	V_{IL}	—	0.8	V
I_{Load} = 1.6mA					
I/O Ports, Three-state Leakage	PA0-7, PC0-7, PD0-5, PG0-7,	I_{OZ}	—	±10	µA
$V_{in} = V_{IH}$ or V_{IL}	PH0-7, PJ0-3, MODA/$\overline{LIR}$, $\overline{RESET}$				
Input Current	All Inputs	I_{in}	—	±1.0	µA
$V_{in} = V_{DD}$ or V_{SS}					
RAM Standby Voltage	Powerdown	V_{SB}	2.0	V_{DD}	V
RAM Standby Current	Powerdown	I_{SB}	—	20	µA
Total Supply Current (See Note 2)		I_{DD}			
RUN:					
Single-Chip Mode	I_{DD}(run)		—	30	mA
WAIT: (All Peripheral Functions Shut Down)					
Single-Chip Mode	I_{DD}(wait)		—	15	mA
STOP: (No Clocks)					
Single-Chip Mode	I_{DD}(stop)		—	100	µA
Input Capacitance	$\overline{IRQ}$, $\overline{XIRQ}$, EXTAL	C_{in}	—	12	pF
	MODA/$\overline{LIR}$, $\overline{RESET}$, all Ports except Port E	C_{in}	—	12	pF

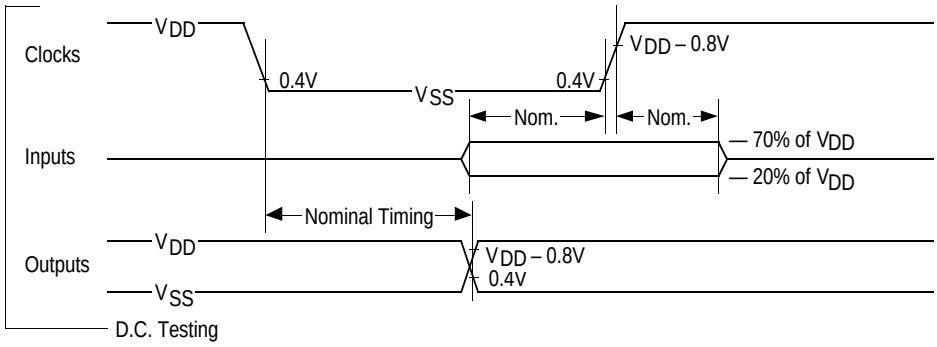
NOTES:

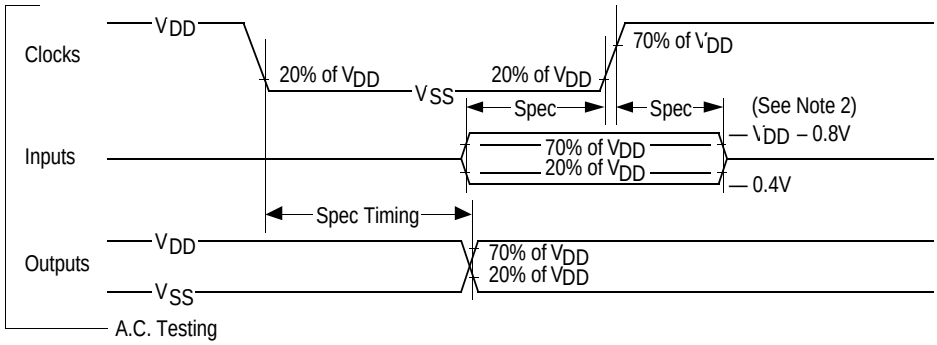
1. V_{OH} specification for $\overline{RESET}$ and MODA is not applicable because they are open-drain pins.
   V_{OH} specification is not applicable to ports C and D in wired-OR mode.
2. All ports configured as inputs,
   $V_{IL} \leq 0.2V$,
   $V_{IL} \geq V_{DD} - 0.2V$,
   No dc loads,
   EXTAL is driven with a square wave, and
   t_{cyc} = 476.5ns.



Equivalent Test Load [1]

Pins	R1	R2	C1
PA0-7, PB0-7, PC0-7, PD0, PD5, PF0-7, PG0-7, PH0-7, PJ0-3, E, R/$\overline{W}$	3.26kΩ	2.38kΩ	90pF
PD1-PD4	3.26kΩ	2.38kΩ	200pF





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 0.4 volts and VDD – 0.8 volts while timing measurements are taken at the 20% and 70% of VDD points.

Figure 13-1. Test Methods

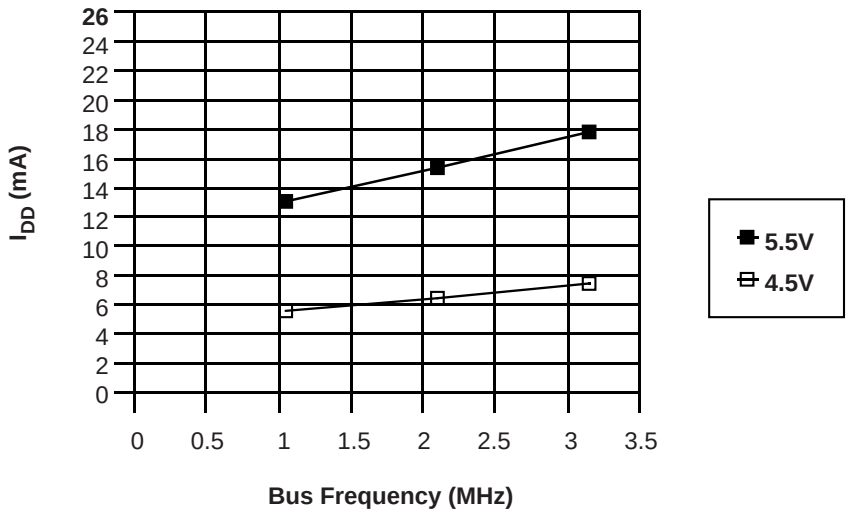


Figure 13-2. Run I_{DD} vs Bus Frequency (Single Chip Mode – 4.5V, 5.5V)

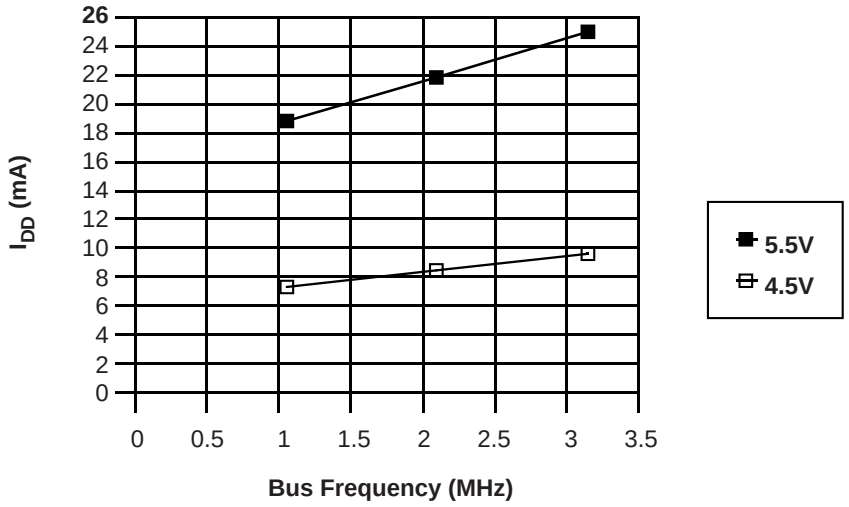


Figure 13-3. Run I_{DD} vs Bus Frequency (Expanded Mode – 4.5V, 5.5V)

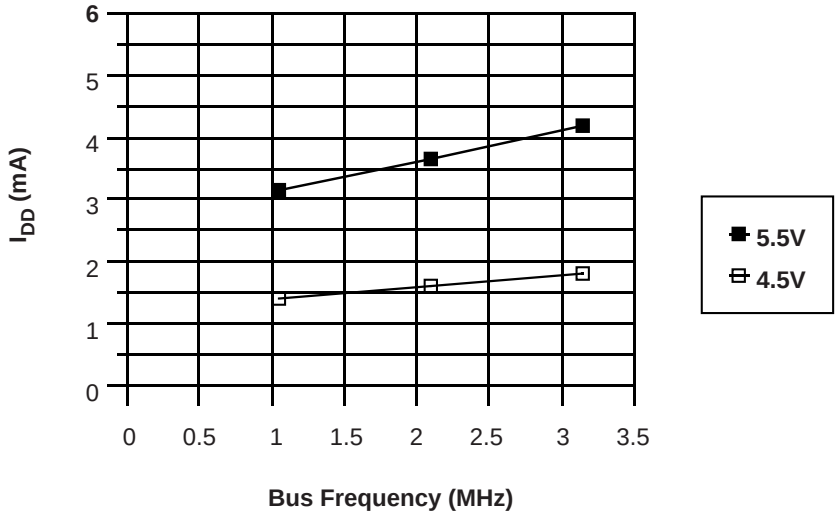


Figure 13-4. Wait I_{DD} vs Bus Frequency (Single Chip Mode – 4.5V, 5.5V)

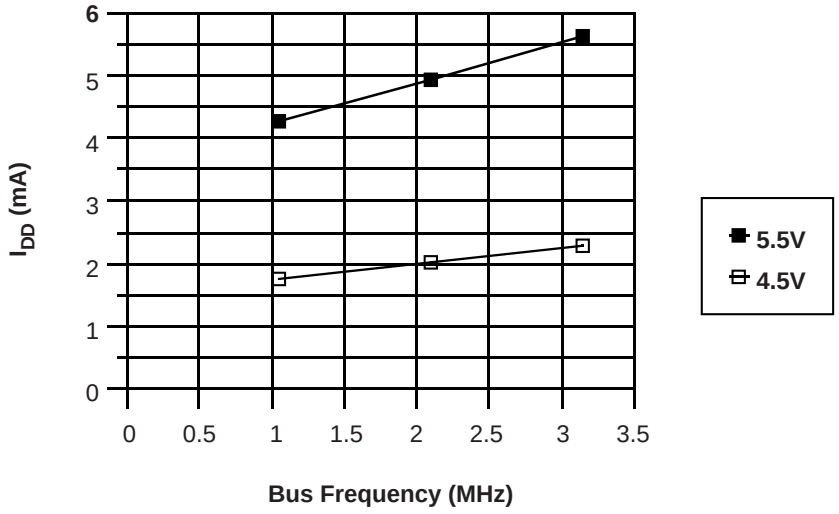


Figure 13-5. Wait I_{DD} vs Bus Frequency (Expanded Mode – 4.5V, 5.5V)

13.5 CONTROL TIMING

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

Characteristic		Symbol	Min	Max	Unit
External Oscillator Frequency	Crystal option	f_{XTAL}	—	8.4	MHz
	External clock option	$4f_o$	DC	8.4	MHZ
Internal Operating Frequency	Crystal (f_{XTAL}/4)	f_o	—	2.1	MHz
	External clock	f_o	DC	2.1	MHZ
Cycle Time	All outputs except XTAL	t_{cyc}	476	—	ns
Crystal Oscillator Startup Time		t_{OXOV}	—	100	ms
Stop Recovery Startup Time	DLY = 0	t_{SRS}	—	4	t_{cyc}
	DLY = 1	t_{SRS}	—	4064	t_{cyc}
Wait Recovery Startup Time		t_{WRS}	—	4	t_{cyc}
Reset Input Pulse Width (See Note 1)		t_{RLRH}			t_{cyc}
	(To guarantee external reset vector)		8	—	
	(Minimum input time; may be pre-empted by internal reset)		1	—	
Mode Programming	Setup time	t_{MPS}	2	—	t_{cyc}
	Hold time	t_{MPH}	0	—	ns
Interrupt Pulse Width, $\overline{IRQ}$ Edge Sensitive Mode	$t_{ILIH} = t_{cyc} + 20ns$	t_{ILIH}	496	—	ns
Interrupt Pulse Period		t_{ILIL}	Note 2	—	t_{cyc}
Processor Control					
$\overline{RESET}$, WAIT, $\overline{IRQ}$	($t_{PCS} = 1/4\ t_{cyc} - 50ns$)	t_{PCS}	69	—	ns
MRDY		t_{PCS}	50	—	ns
HALT	($t_{PCS} = 1/4\ t_{cyc} + 20ns$)	t_{PCS}	170	—	ns
Bus Tri State Enable	($t_{TSE} = 1/4\ t_{cyc} + 40ns$)	t_{TSE}	—	159	ns
Bus Tri State Disable		t_{TSD}	—	65	ns

NOTES:

1. $\overline{RESET}$ will be recognised during the first clock cycle it is held low. Internal circuitry then drives the pin low for four clock cycles, releases the pin, and samples the pin level two cycles later to determine the source of the interrupt.
2. The minimum period t_{ILIL} should not be less than the number of cycles it takes to execute the interrupt service routine plus 21 t_{cyc}.
3. All timing is shown with respect to 20% V_{DD} and 70% V_{DD} unless otherwise noted.

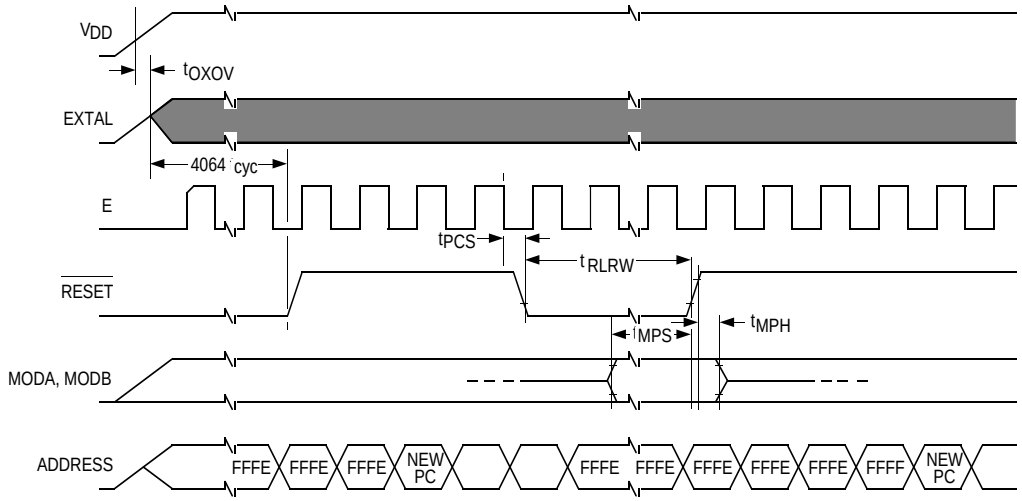


Figure 13-6. POR External RESET Timing Diagram

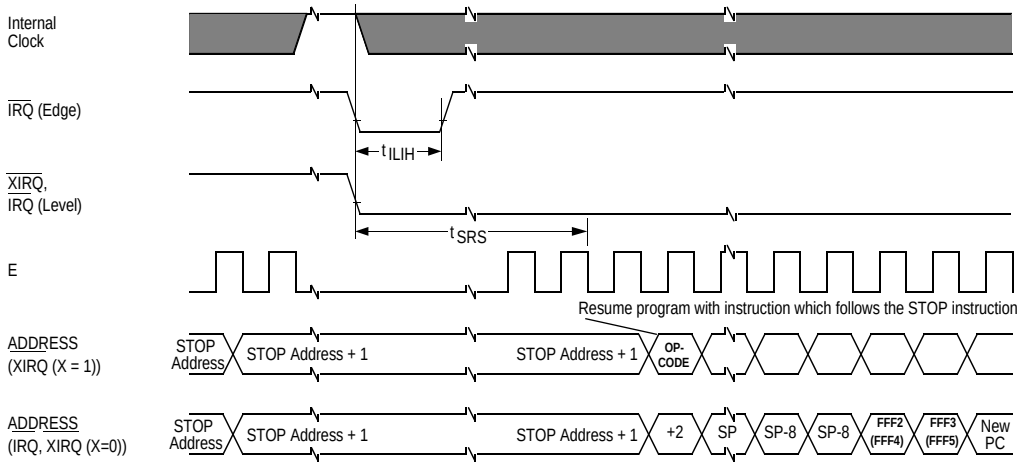


Figure 13-7. STOP Recovery Timing Diagram

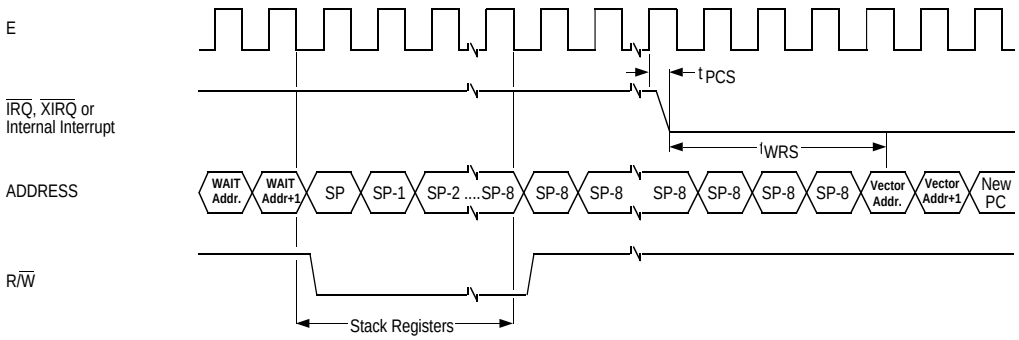


Figure 13-8. WAIT Recovery from Interrupt Timing Diagram

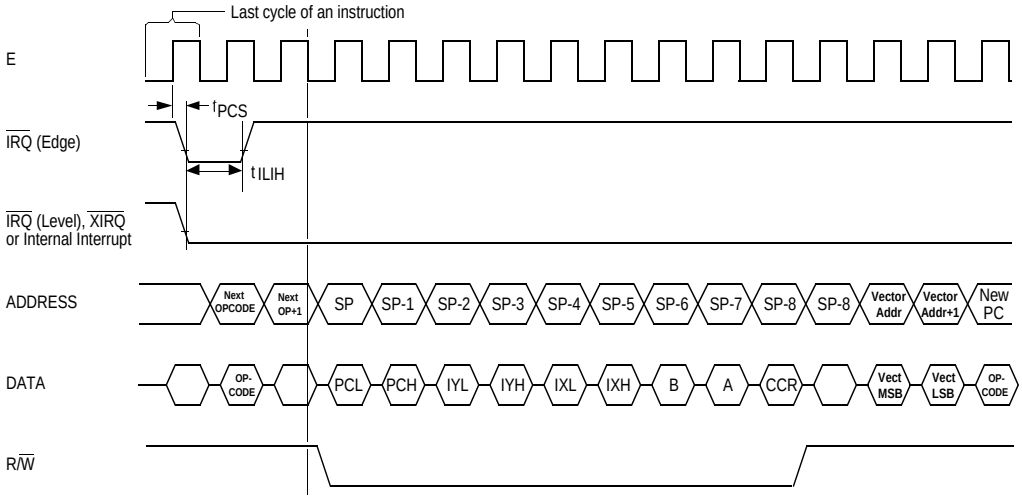


Figure 13-9. Interrupt Timing Diagram

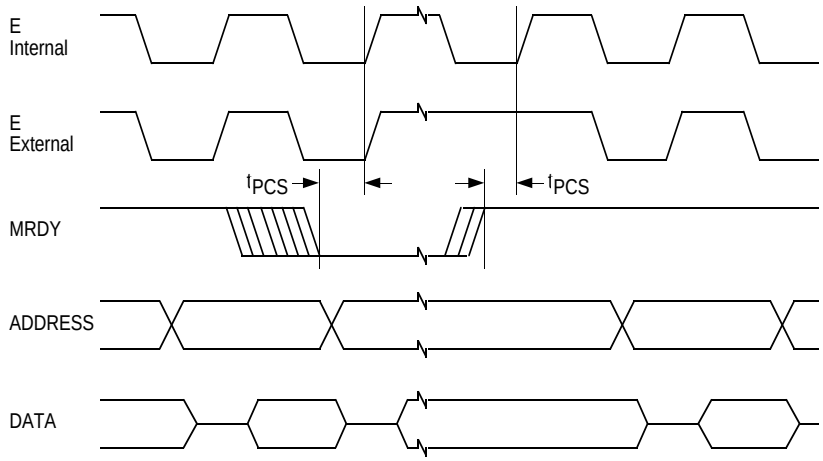


Figure 13-10. Memory Ready Timing Diagram

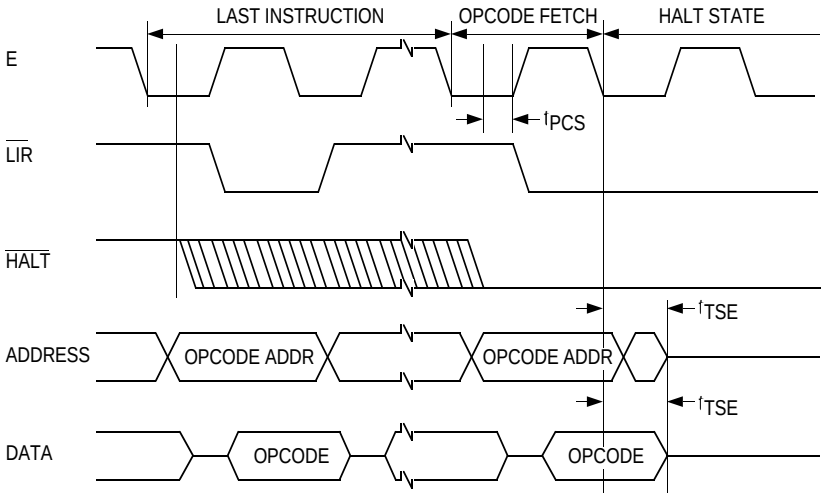


Figure 13-11. Entering HALT

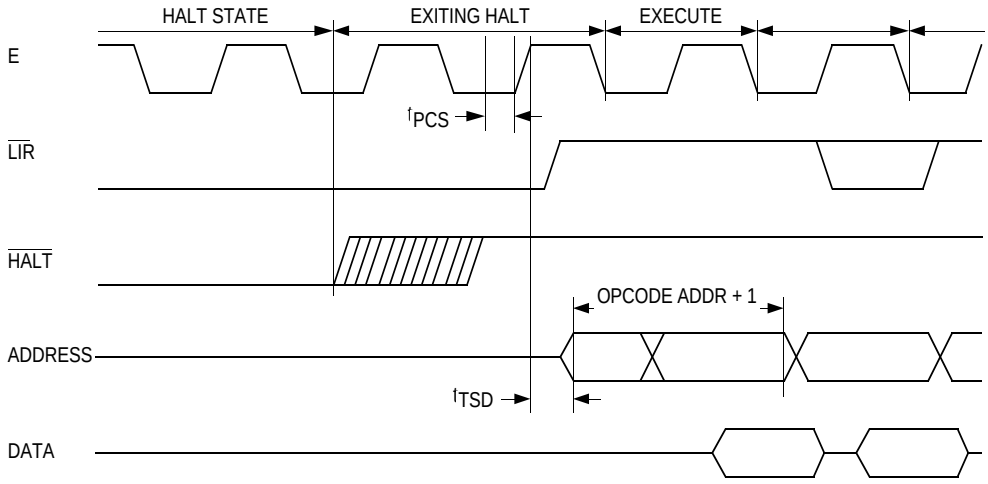


Figure 13-12. Exiting HALT

13.6 PERIPHERAL PORT CHARACTERISTICS

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

Characteristic	Symbol	Min	Max	Unit
Frequency of Operation (E Clock)	f_o	—	2.1	MHz
E Clock Period	t_{cyc}	476	—	ns
Peripheral Data Setup Time				
Port A, C, D, J	t_{PDSU}	100	—	ns
Port E	t_{PDSU}	100	—	ns
Port G, H (t_{PDSU} = -1/4 t_{cyc} + 93ns)	t_{PDSU}	– 26	—	ns
Peripheral Data Hold Time				
Port A, C, D, J	t_{PDH}	80	—	ns
Port E	t_{PDH}	80	—	ns
Port G, H (t_{PDH} = 1/4 t_{cyc} + 130ns)	t_{PDH}	249	—	ns
Delay Time, Peripheral Data Write				
Port J1, J2, A3, A4, A5, A6, A7	t_{PWD}	—	150	ns
Port B, C, D, F, G, H, A0, A1, A2, J0, J3 (t_{PWD} = 1/4 t_{cyc} + 90ns)	t_{PWD}	—	209	ns

NOTES:

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

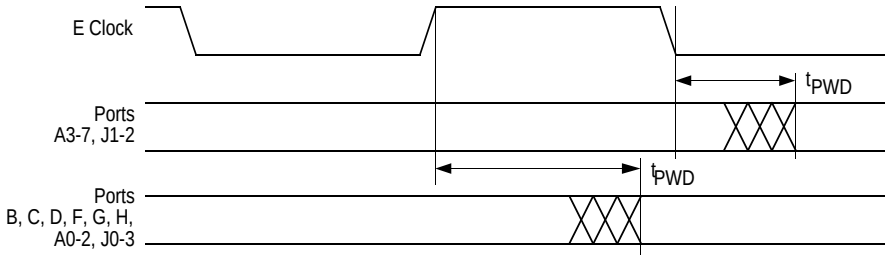


Figure 13-13. Port Write Timing Diagram

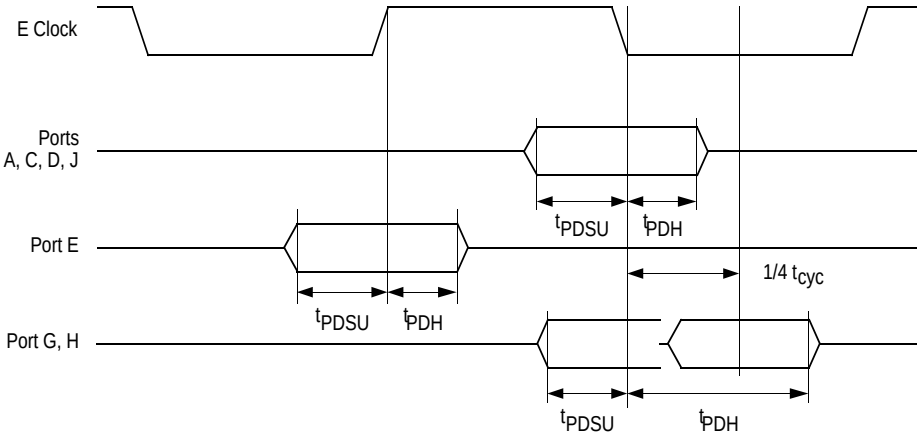


Figure 13-14. Port Read Timing Diagram

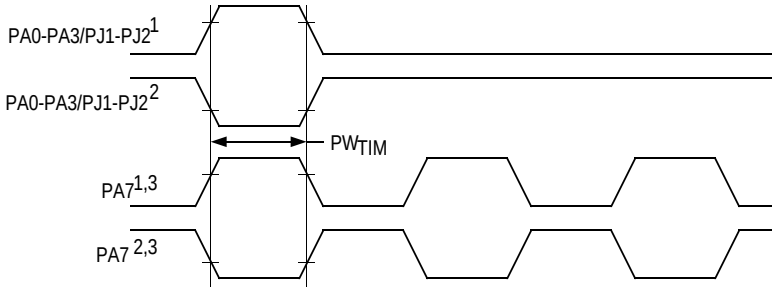
13.7 TIMER CHARACTERISTICS

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

Characteristic	Symbol	Min	Max	Unit
Timer Pulse Width Input Capture Pulse Accumulator Input ($PW_{TIM} = t_{cyc}$ + 20ns)	PW_{TIM}	496	—	ns
Timer Output Compare High Valid	t_{OCH}	—	310	ns
Timer Output Compare Low Valid	t_{OCL}	—	295	ns
Timer Input Capture Response Delay Min = t_{cyc} + 20ns Max = 2 t_{cyc} + 220ns	t_{CAP}	496	1172	ns

NOTES:

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



Notes:
1. Rising edge sensitive input.
2. Failing edge sensitive input.
3. Maximum pulse accumulator clocking rate is E frequency divided by 2.

Figure 13-15. Timer Inputs Timing Diagram

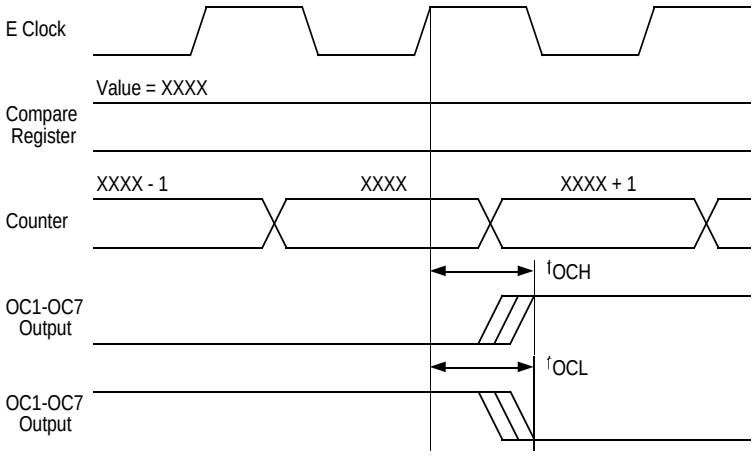


Figure 13-16. Output Compare Timing Diagram

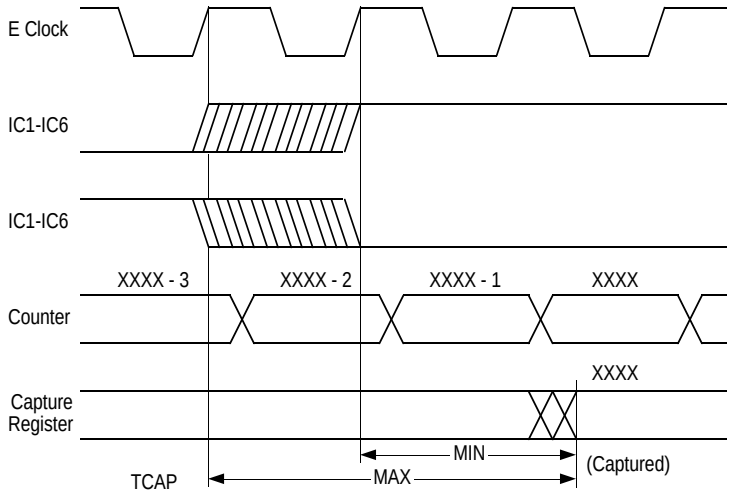


Figure 13-17. Input Capture Timing Diagram

13.8 A/D CONVERTER CHARACTERISTICS

(V_{DD} = 5.0 Vdc ± 10%, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H, E = 750kHz to 2.1MHz unless otherwise noted)

Characteristic	Parameter	Min	Absolute	Max	Unit
Resolution	Number of bits resolved by the A/D	10	—	—	Bits
Nonlinearity	Maximum deviation from the ideal A/D transfer characteristics	—	—	TBD	LSB
Zero Error	Difference between the output of an ideal and an actual A/D for zero input voltage	—	—	TBD	LSB
Full Scale Error	Difference between the output of an ideal and an actual A/D for full scale input voltage	—	—	TBD	LSB
Total Unadjusted Error	Maximum sum of Nonlinearity, zero error and full scale error	—	—	TBD	LSB
Quantization Error	Uncertainty due to converter resolution	—	—	±0.5	LSB
Absolute Accuracy (See Note 1)	Difference between the actual input voltage and the full scale weighted equivalent of the binary output code, all error sources included	—	—	TBD	LSB
Conversion Range (See Notes 3 and 4)	Analog input voltage range	V_{rl}	—	V_{rh}	V
V_{rh} (See Note 2)	Maximum analog reference voltage	V_{rl}	—	V_{DD} + 0.1	V
V_{rl} (See Note 2)	Minimum analog reference voltage	V_{SS} - 0.1	—	V_{rh}	V
Delta V_r (See Note 2)	Minimum difference between V_{rh} and V_{rl}	0	—	—	V
Conversion Time	Total time to perform a single analog to digital conversion: 1. E Clock 2. Internal RC Oscillator	 — —	 36 —	 — t_{cyc} + 24	 t_{cyc} µs
Monotonicity	Conversion result never decreases with an increase in input voltage and has no missing codes	Guaranteed			
Zero Input Reading (See Note 3)	Conversion result when $V_{in} = V_{rl}$	$000	—	—	Hex
Full Scale reading (See Note 4)	Conversion result when $V_{in} = V_{rh}$	—	—	$3FF	Hex
Sample Acquisition Time	Analog Input Acquisition sampling time: 1. E Clock 2. Internal RC Oscillator	 — —	 13 —	 — 8.7	 t_{cyc} µs
Sample Hold Capacitance	Input capacitance during sample PE0-PE7	—	35 (Typ)	—	pF
Input Leakage	Input leakage on A/D pins: PE0-PE7 V_{rl}, V_{rh}	 — —	 — —	 100 250	 nA µA

NOTES:

1. Source impedances greater then 10kΩ will adversely affect accuracy, due mainly to input leakage.
2. Performance verified down to 2.5V Delta V_r, but accuracy is tested and guaranteed at Delta V_r = 5V ± 10%.
3. Minimum analog input voltage should not go below V_{SS} - 0.3V.
4. Maximum analog input voltage should not exceed 1.125V_{rh}.

13.9 EXPANSION BUS TIMING

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

Num	Characteristic	Symbol	Min	Max	Unit
	Frequency of Operation (E-Clock)	f_o	—	2.1	MHz
1	Cycle Time	t_{cyc}	476	—	ns
2	Pulse Width E Low (1/2 t_{cyc} - 23ns)	t_{ELEH}	215	—	ns
3	Pulse Width E High (1/2 t_{cyc} - 28ns)	t_{EHEL}	210	—	ns
4	E Rise and Fall Time	t_R, t_F	—	20	ns
9	Address Hold Time (t_{AH} = 1/8 t_{cyc} - 29.5ns) (See Note 1(A))	t_{AH}	30	—	ns
12	Address Valid Time to E Rise (See Note 1(B))	t_{AV}	120	—	ns
17	Read Data Setup Time	t_{DSR}	30	—	ns
18	Read Data Hold Time	t_{DHR}	10	—	ns
19	Write Data Delay Time	t_{DDW}	—	80	ns
21	Write Data Hold Time	t_{DHW}	50	—	ns
29	MPU Address Access Time ($t_{ACCA} = t_{AV} + t_R + t_{EHEL} - t_{DSR}$ (See Note 1(A))	t_{ACCA}	320	—	ns

NOTES:

1. Input clock with duty cycle other than 50% will affect the bus performance. Timing parameters affected by the input clock duty cycle are identified by (A) and (B). To re-calculate approximate bus timing values, substitute the following expressions in place of 1/8 t_{cyc} in the above formulae where applicable:

   (A) (1-DC) x 1/4 t_{cyc}

   (B) DC x 1/4 t_{cyc}

   where DC is the decimal value of duty cycle percentage (High time).

2. All timing is shown with respect to 20% V_{DD} and 70% V_{DD}.

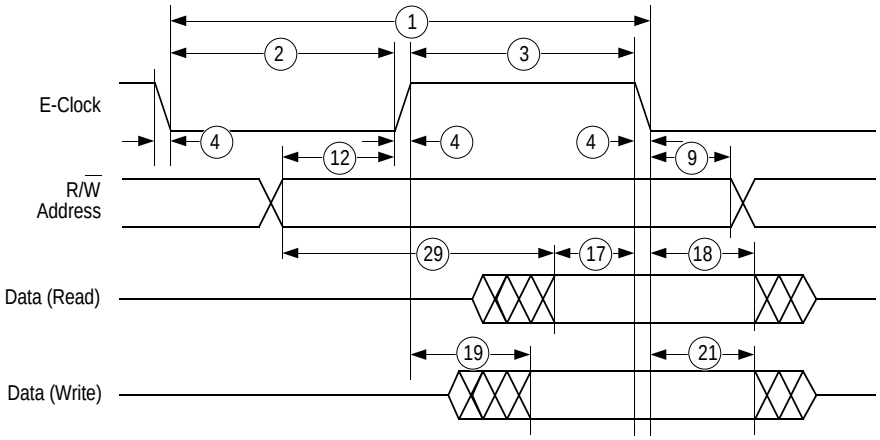


Figure 13-18. Non-multiplexed Expanded Bus

13.10 SERIAL PERIPHERAL INTERFACE (SPI) TIMING

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

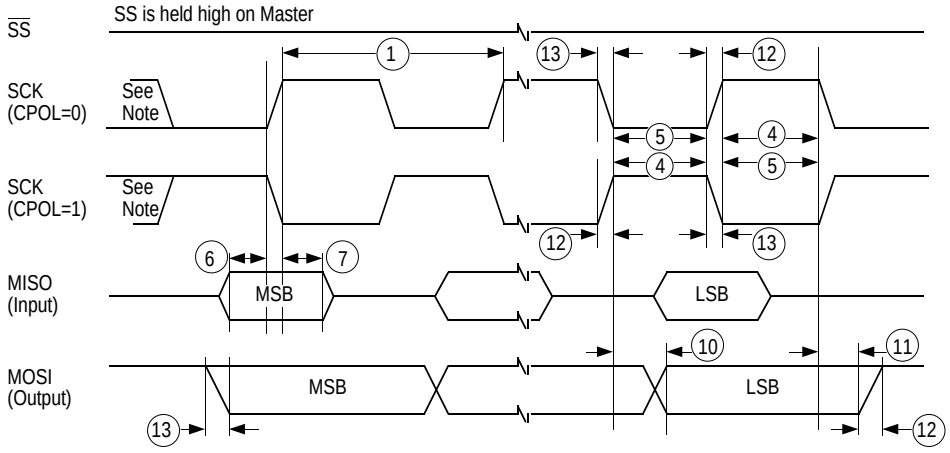
Num	Characteristic		Symbol	Min	Max	Unit
	Operating Frequency	Master Slave	$f_{op(m)}$ $f_{op(s)}$	dc dc	0.5 2.1	f_o MHz
1	Cycle Time	Master Slave	$t_{cyc(m)}$ $t_{cyc(s)}$	2.0 480	— —	t_{cyc} ns
2	Enable Lead Time	Master Slave	$t_{lead(m)}$ $t_{lead(s)}$	* 240	— —	ns ns
3	Enable Lag Time	Master Slave	$t_{lag(m)}$ $t_{lag(s)}$	* 240	— —	ns ns
4	Clock (SCK) High Time	Master Slave	$t_{w(SCKH)m}$ $t_{w(SCKH)s}$	340 190	— —	ns ns
5	Clock (SCK) Low Time	Master Slave	$t_{w(SCKL)m}$ $t_{w(SCKL)s}$	340 190	— —	ns ns
6	Data Setup Time	Master Slave	$t_{su(m)}$ $t_{su(s)}$	100 100	— —	ns ns
7	Data Hold Time	Master Slave	$t_{h(m)}$ $t_{h(s)}$	100 100	— —	ns ns
8	Access Time (Time to Data Active from High-Impedance State)	Slave	t_a	0	120	ns
9	Disable Time (Hold Time to High-Impedance State)	Slave	t_{dis}	—	240	ns
10	Data Valid (After Enable Edge)**		$t_{v(s)}$	—	240	ns
11	Data Hold Time (Outputs) (After Enable Edge)		t_{ho}	- 40	—	ns
12	Rise Time (20% V_{DD} to 70% V_{DD}, C_L = 200pF) SPI Outputs (SCK, MOSI, MISO) SPI Inputs (SCK, MOSI, MISO, $\overline{SS}$)		 $t_{r(m)}$ $t_{r(s)}$	 — —	 100 2.0	 ns µs
13	Fall Time (70% V_{DD} to 20% V_{DD}, C_L = 200pF) SPI Outputs (SCK, MOSI, MISO) SPI Inputs (SCK, MOSI, MISO, $\overline{SS}$)		 $t_{f(m)}$ $t_{f(s)}$	 — —	 100 2.0	 ns µs

NOTES:

* Signal production depends on software.

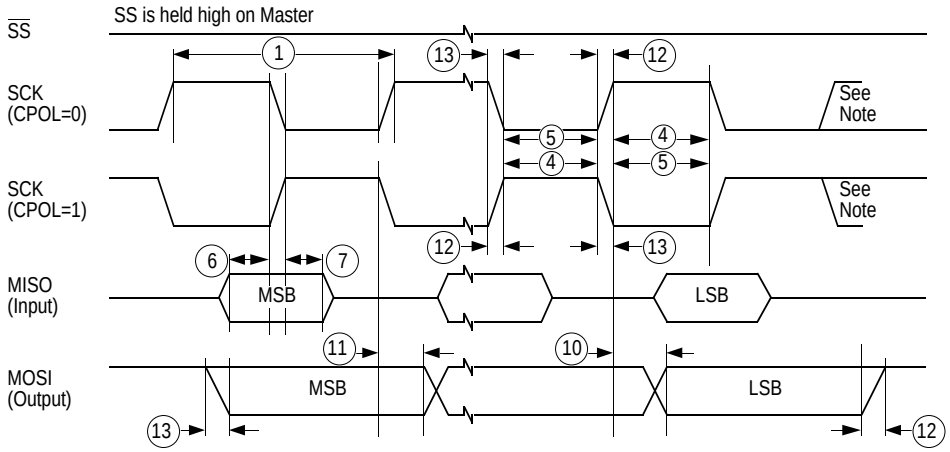
** Assumes 200 pF load on all SPI pins.

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



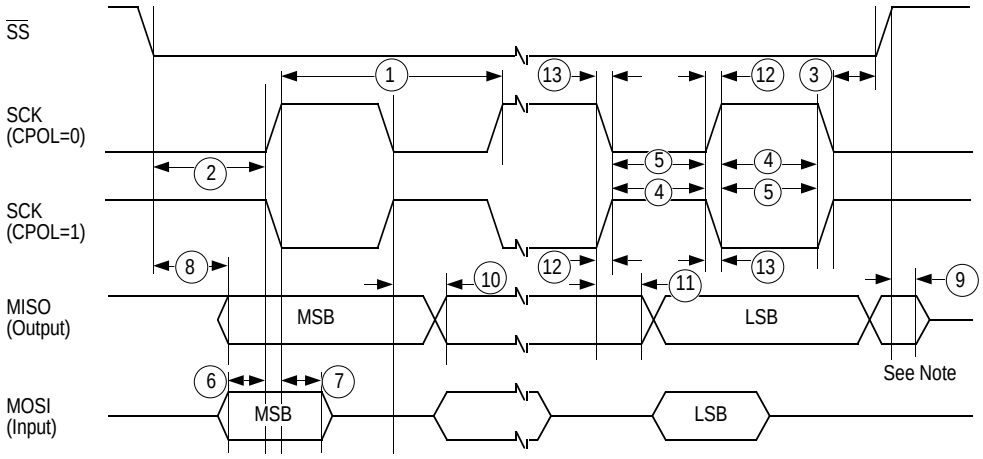
Note: This first edge is generated internally but is not seen at the SCK pin.

Figure 13-19. SPI Master Timing (CPHA = 0)



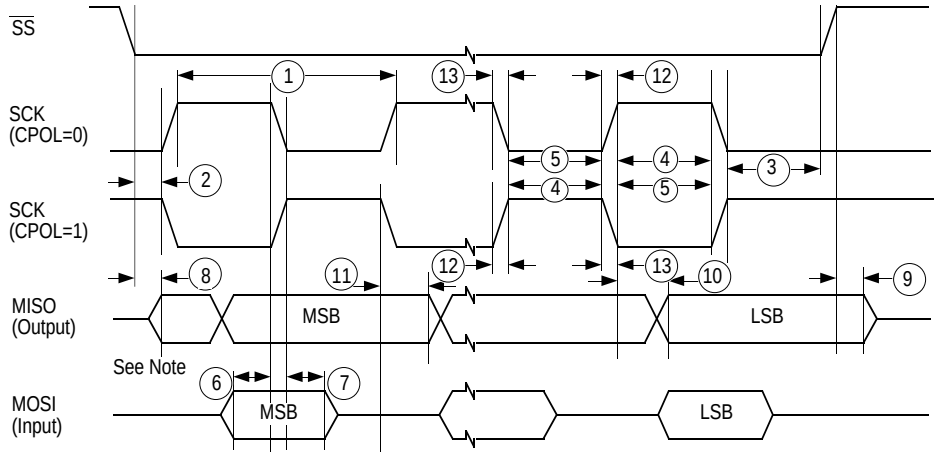
Note: This last edge is generated internally but is not seen at the SCK pin.

Figure 13-20. SPI Master Timing (CPHA = 1)



Note: Not defined but normally MSB of character just received.

Figure 13-21. SPI Slave Timing (CPHA = 0)



Note: Not defined but normally LSB of character previously transmitted.

Figure 13-22. SPI Slave Timing (CPHA = 1)

13.11 EVENT COUNTER CHARACTERISTICS

Table 13.1. Clock Input (Required Limit)

Clock Input (Required Limit)

Num	Characteristic	Symbol	Min	Max	Unit
1	Cycle Time of External Clock Input	t_{cycex}	$3xt_{cyc}$	—	ns
2	Clock High Time	t_{wckh}	$1.5xt_{cyc}$	—	ns
3	Clock Low Time	t_{wckl}	$1xt_{cyc}$	—	ns
4	Rise Time	t_{rck}	—	$0.25xt_{cyc}$	ns
5	Fall Time	t_{fck}	—	$0.25xt_{cyc}$	ns

Table 13.2. Clock Input (Guaranteed Limit)

Clock Input (Guaranteed Limit)

Num	Characteristic	Symbol	Min	Max	Unit
6	Propagation Delay External Clock Rise to EVO Valid	t_{ovr}	$1.5xt_{cyc}$	$2.5xt_{cyc}$ + 240	ns
7	Propagation Delay External Clock Fall to EVO Valid	t_{ovf}	$1.5xt_{cyc}$	$2.5xt_{cyc}$ + 240	ns

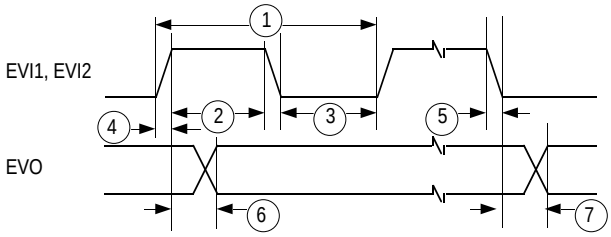


Figure 13-23. Event Counter Mode 1, 2, 3 – Clock Input Timing Diagram

Table 13.3. Clock Gate Input (Guaranteed Limit)

Clock Gate Input (Required Limit)

Num	Characteristic	Symbol	Min	Max	Unit
1	Gate Input Setup Time to E Fall	t_{gsu}	$0.25xt_{cyc}$	—	ns
2	Gate Input Hold Time to E Fall	t_{gh}	$0.5xt_{cyc}$	—	ns

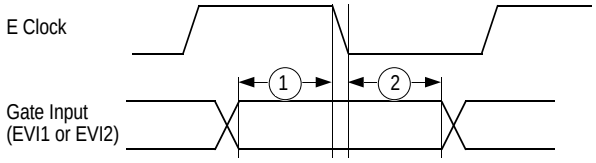


Figure 13-24. Event Counter Mode 1, 2, 3 – Clock Gate Input Timing Diagram

This page intentionally left blank

SECTION 14

MECHANICAL DATA

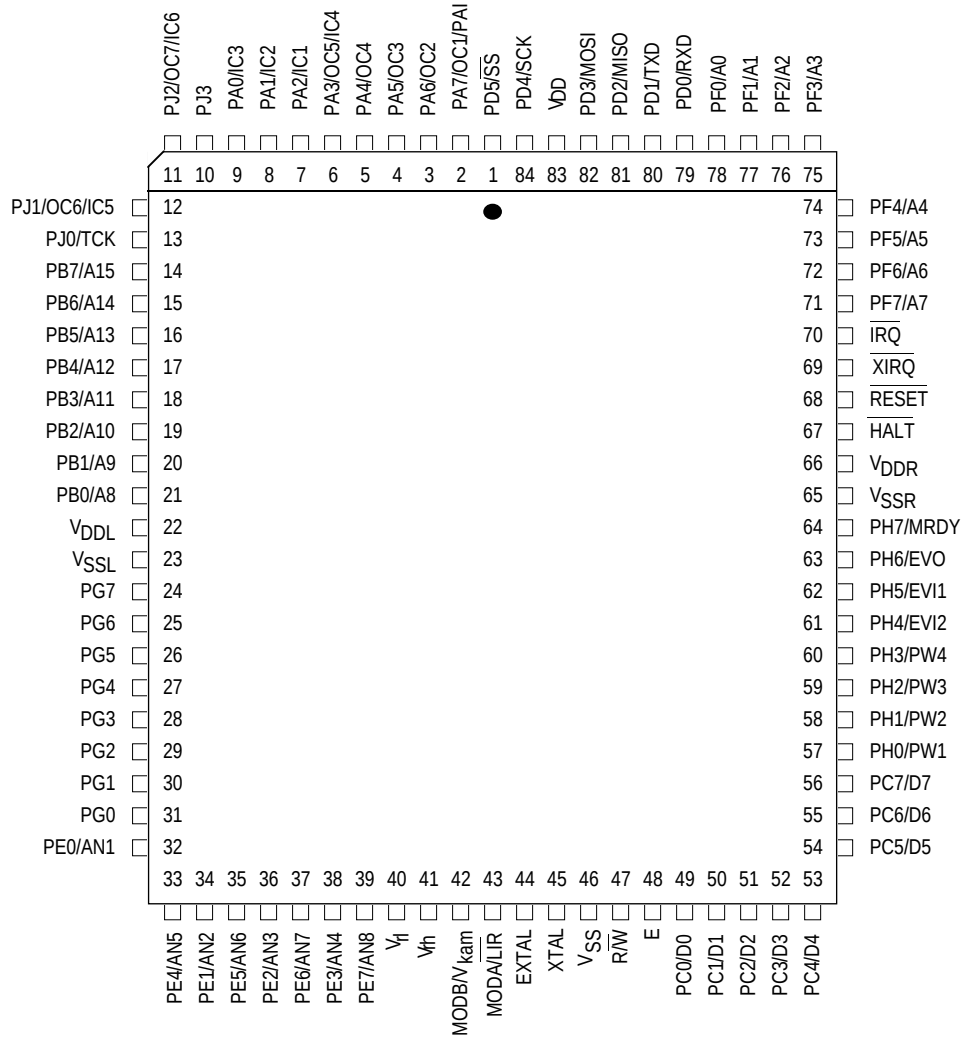
This section contains the pin assignments, packaging dimensions and ordering information for the MC68HC11G5 MCU.

14.1 ORDERING INFORMATION

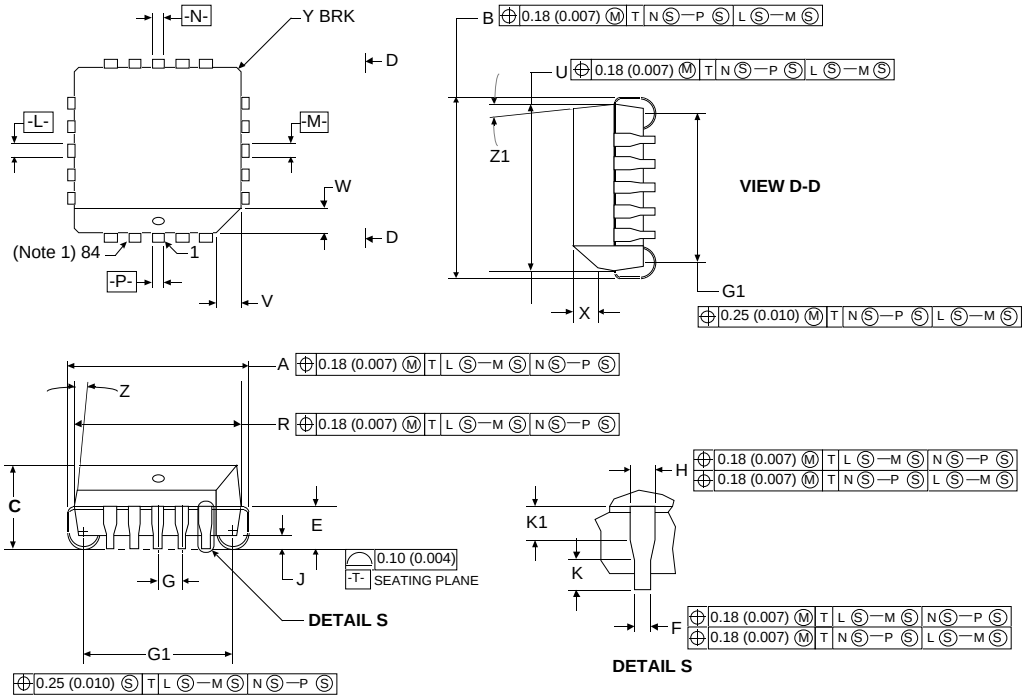
Package Type	Temperature	Part Number
PLCC (CFN Suffix)	– 40° to 85°C	MC68HC11G5CFN

14.2 PIN ASSIGNMENTS

The MC68HC11G5 is offered in an 84-pin PLCC (Quad Surface Mount plastic) package.



14.3 PACKAGE DIMENSIONS



DIM	MILLIMETERS		INCHES	
	MIN	MAX	MIN	MAX
A	30.10	30.35	1.185	1.195
B	30.10	30.35	1.185	1.195
C	4.20	4.57	0.165	0.180
E	2.29	2.79	0.090	0.110
F	0.33	0.48	0.013	0.019
G	1.27 BSC		0.050 BSC	
H	0.66	0.81	0.026	0.032
J	0.51	—	0.020	—
K	0.64	—	0.025	—
R	29.21	29.36	1.150	1.156
U	29.21	29.36	1.150	1.156
V	1.07	1.21	0.042	0.048
W	1.07	1.21	0.042	0.056
X	1.07	1.42	0.042	0.056
Y	—	0.50	—	0.020
Z	2°	10°	2°	10°
G1	28.20	28.70	1.110	1.130
K1	1.02	—	0.040	—
Z1	2°	10°	2°	10°

NOTES:

1. DUE TO SPACE LIMITATION, CASE 780-01 SHALL BE REPRESENTED BY A GENERAL CASE OUTLINE DRAWING.
2. DATUMS -L-, -M-, -N-, AND -P- DETERMINED WHERE TOP OF LEAD SHOULDER EXIT PLASTIC BODY AT MOLD PARTING LINE.
3. DIM G1, TRUE POSITION TO BE MEASURED AT DATUM -T-, SEATING PLANE
4. DIM R AND U DO NOT INCLUDE MOLD PROTRUSION. ALLOWABLE MOLD PROTRUSION IS 0.25 (0.010) PER SIDE
5. DIMENSIONS AND TOLERANCING PER ANSI Y14.5M, 1982
6. CONTROLLING DIMENSION: INCH

This page intentionally left blank

Literature Distribution Centres:
USA: Motorola Literature Distribution; P.O. Box 20912; Phoenix, Arizona 85036.
EUROPE: Motorola Ltd.; European Literature Centre; 88 Tanners Drive, Blakelands, Milton Keynes, MK14 5BP, England.
ASIA PACIFIC: Motorola Semiconductors (H.K.) Ltd.; Silicon Harbour Center, No. 2, Dai King Street, Tai Po Industrial Estate, Tai Po, N.T., Hong Kong.
JAPAN: Nippon Motorola Ltd.; 4-32-1, Nishi-Gotanda, Shinagawa-ku, Tokyo 141, Japan.