HC05G3GRS/D
REV 1.1

68HC05G3
68HC705G4

SPECIFICATION
(General Release)

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

SECTION 1 INTRODUCTION

1.1 GENERAL

The MC68HC05G3 (705G4) is an 80-pin microcontroller unit (MCU) with highly sophisticated on-chip peripheral functions. The memory map of MC68HC05G3 (ROM device) includes 24 Kbytes of user ROM and 768 bytes of RAM. The memory map of MC68HC705G4 erasable programmable read-only memory (EPROM device) includes 32 Kbytes of user EPROM and 1 Kbyte of RAM. The MCU has nine ports: A, B, C, D, E, F, G, H, and J. Ports A, C, D, E, G, and H each have eight input-output (I/O) pins, ports B and F each have eight input-only pins, and port J has four output-only pins. The MC68HC05G3 includes a time-based circuit, 8- and 16-bit timers, an 8-bit pulse width modulator, a computer operating properly (COP) watchdog timer, an 8-bit analog/digital (A/D) converter, eight key wakeup interrupts, and two serial peripheral interfaces.

1.2 FEATURES

- Low Cost
- HC05 Core
- 80-Pin Quad Flat Package (QFP)
- 24,592 Bytes of Mask ROM or 32,784 Bytes of EPROM (Including 16 Bytes of User Vectors)
- 768 Bytes (ROM Device) or 1024 Bytes (EPROM Device) of On-Chip RAM
- 48 Bidirectional I/O Lines, 16 Input-Only Lines, Four Output-Only Lines
- 16-Bit Timer with Output Compare and Input Capture
- 8-Bit Event Counter/Modulus Clock Divider
- COP Watchdog Timer
- Two Serial Peripheral Interfaces (SPI)
- Four Channels of 8-bit Pulse Width Modulator (PWM)
- Eight Channels of 8-Bit A/D Converter
- On-Chip Time-Based Circuits
- Dual Oscillators and Selectable System Clock Frequency
- Power-Saving Stop Mode/Wait Mode
- Time Base Interrupts
- Two IRQ Inputs
- Key Wakeup Interrupt with 8-Bit Inputs

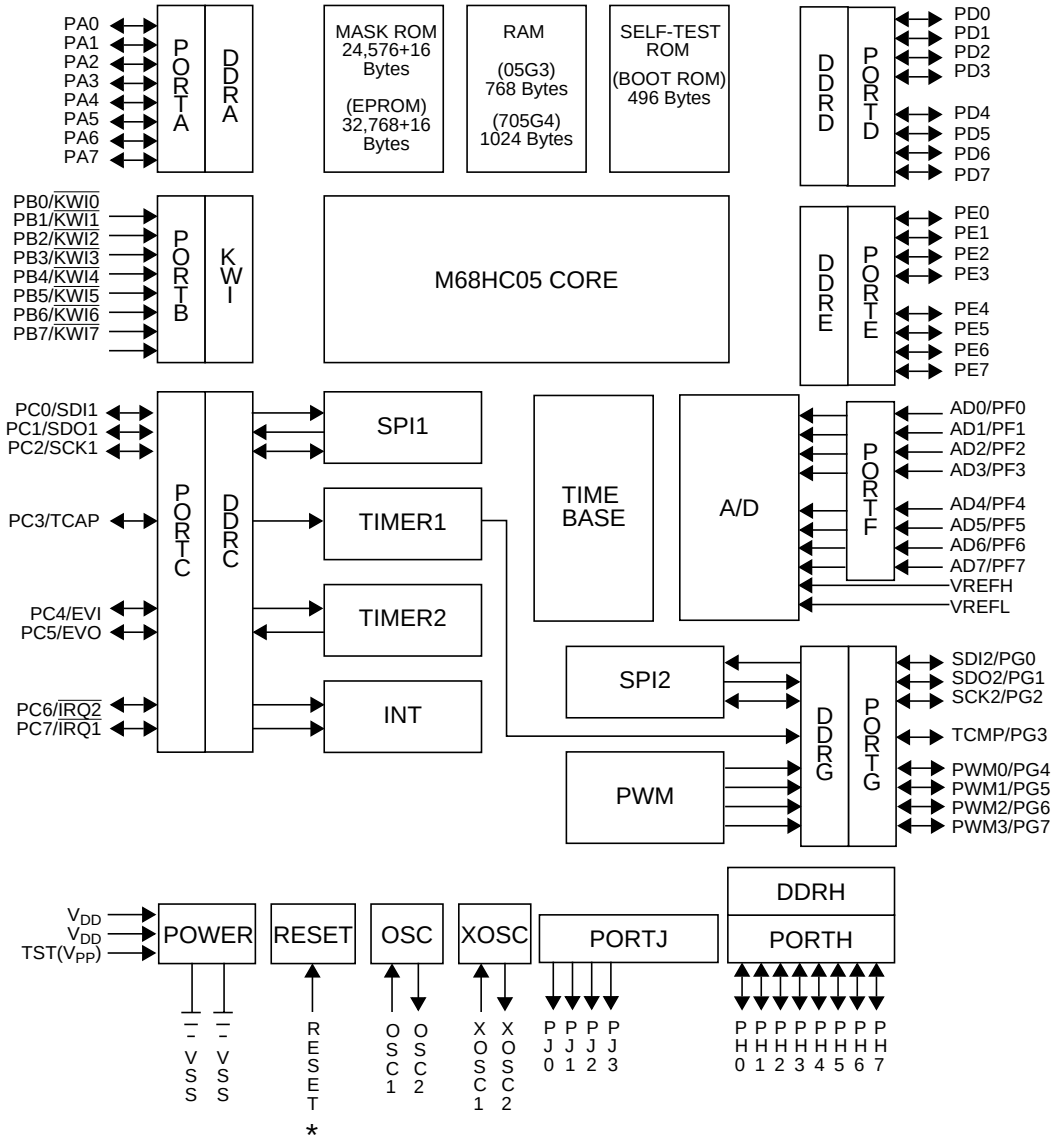


Figure 1-1: Block Diagram of the MC68HC05G3 (705G4)

1.3 MASK OPTIONS

The three mask options on the MC68HC05G3 are: RSTR ($\overline{\text{RESET}}$ pin pullup resistor), OSCR (OSC feedback resistor), and XOSCR (XOSC feedback and damping resistors). The MC68HC705G4 has no mask options.

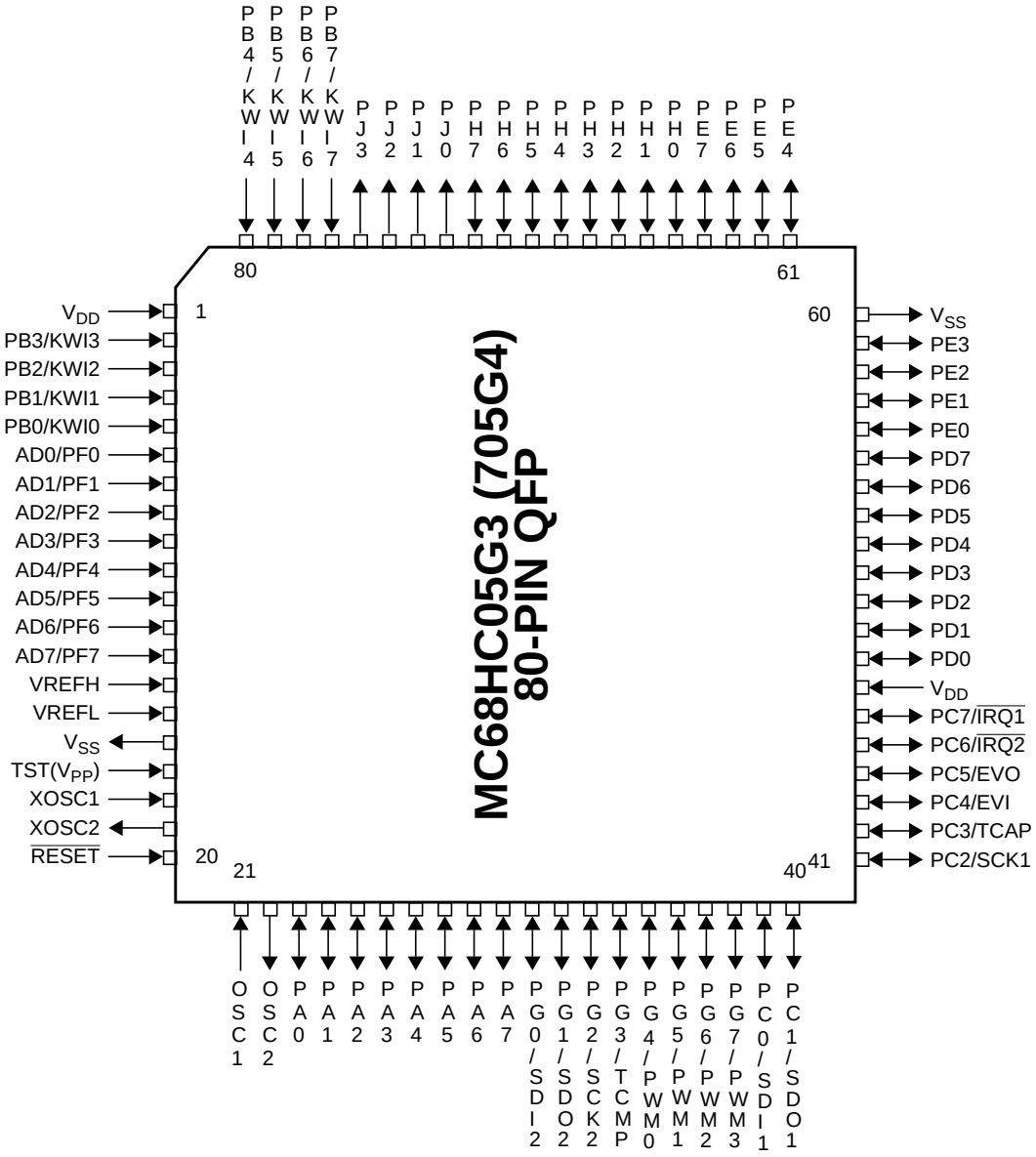


Figure 1-2: Pin Assignment for Single-Chip Mode

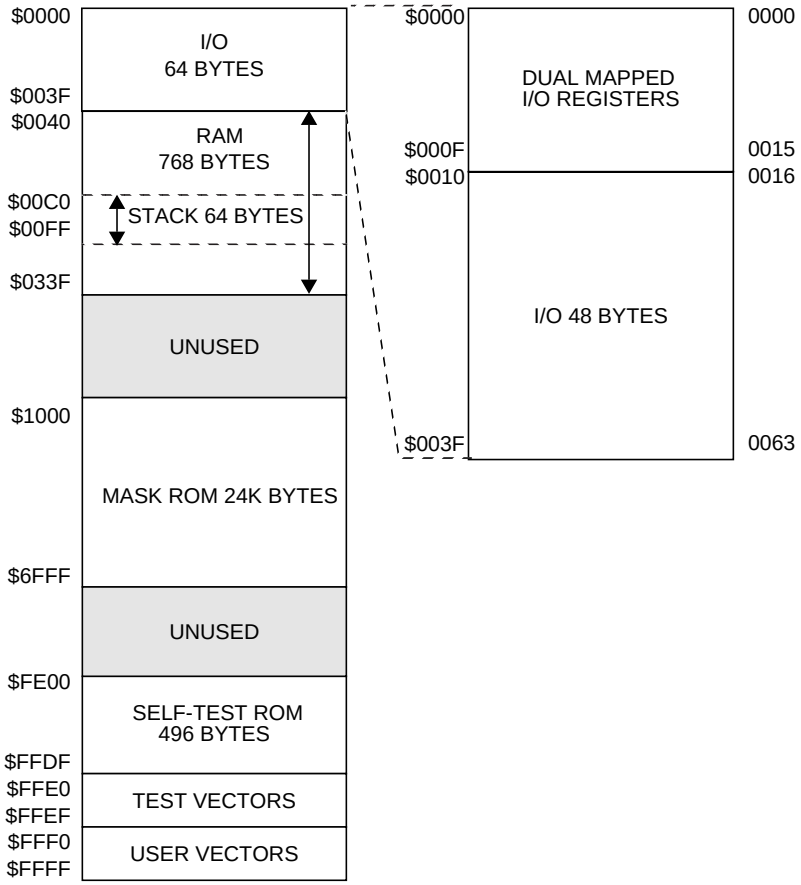


Figure 1-3: Memory Map of MC68HC05G3

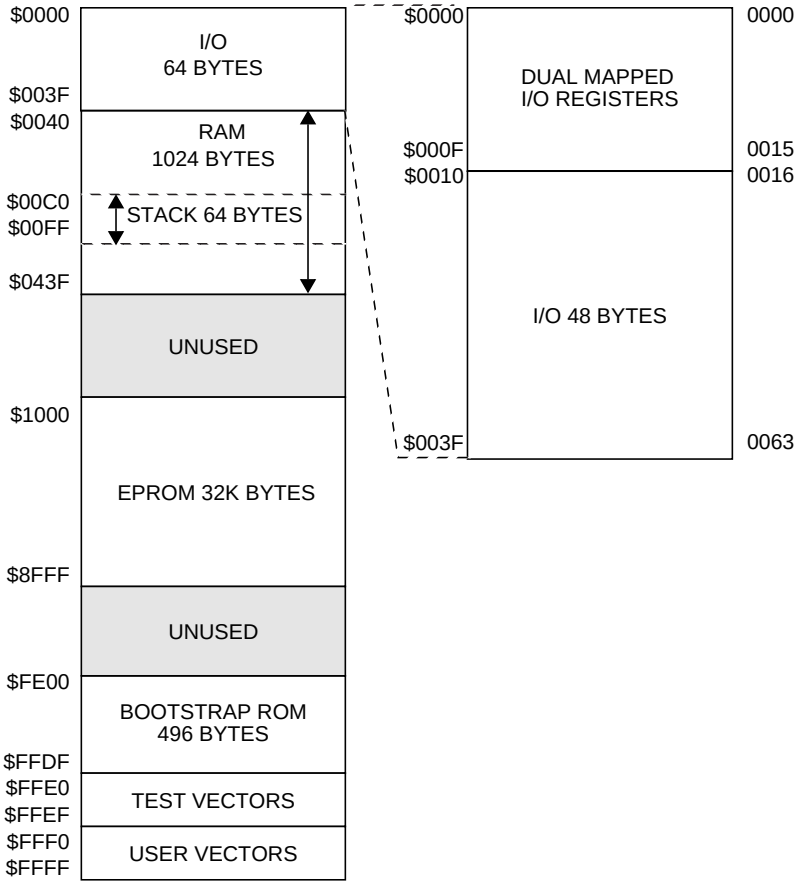


Figure 1-4: Memory Map of MC68HC705G4

1.4 SYSTEM CONFIGURATIONS

The MC68HC05G3 (705G4) has several options. The sections below describe oscillator clocks, time base, and I/O pin configurations.

1.4.1 OSCILLATORS AND CLOCK DISTRIBUTIONS

There are two oscillator blocks: OSC and XOSC. Several combinations of the clock distributions are allowed for the modules in the MC68HC05G3 (705G4). Refer to the following block diagram.

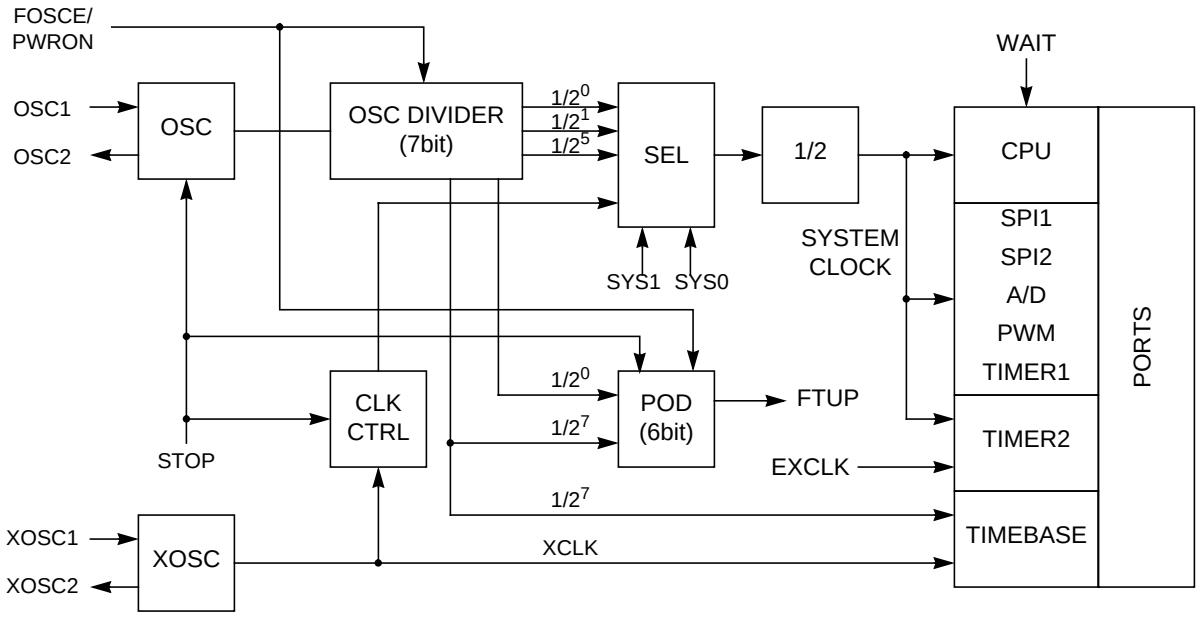


Figure 1-5: Clock Signal Distribution

1.4.1.1 OSC ON LINE

The main oscillator (OSC) can be stopped to conserve power via the STOP instruction or the FOSCE bit in the MISC register. The effects of restarting the OSC will vary depending on the current state of the MCU, including SYS0:1 and FOSCE.

If XOSC is not used, XOSC1 should be connected to either Vss or Vdd.

If OSC is the system clock, FOSCE should remain 1. Executing the STOP instruction in this condition will halt OSC, put the MCU into a low-power mode and clear the 6-bit power-on delay (POD) counter. The 7-bit divider is not initialized. Exiting STOP with external IRQ or reset re-starts the oscillator. When the POD counter overflows, internal reset is released and execution can begin. The stabilization time will vary between 8064 and 8192 counts.

NOTE: Exiting STOP with external reset will always return the MCU to the states defined by the register definitions, such as SYS0:1=0:0, FOSCE=1.

1.4.1.2 XOSC ON LINE

The secondary oscillator (XOSC) runs continuously after power-up.

If XOSC is the system clock (SYS0:1=1:1), OSC can be stopped either by the STOP instruction or by clearing the FOSCE bit.

The sub oscillator (XOSC) never stops except during power down. This clock also may be used as the source for the system clock and/or time base.

OSC and XOSC pins have options for feedback and damping resistor implementations. These options are set through mask option and may be read through the MOSR register.

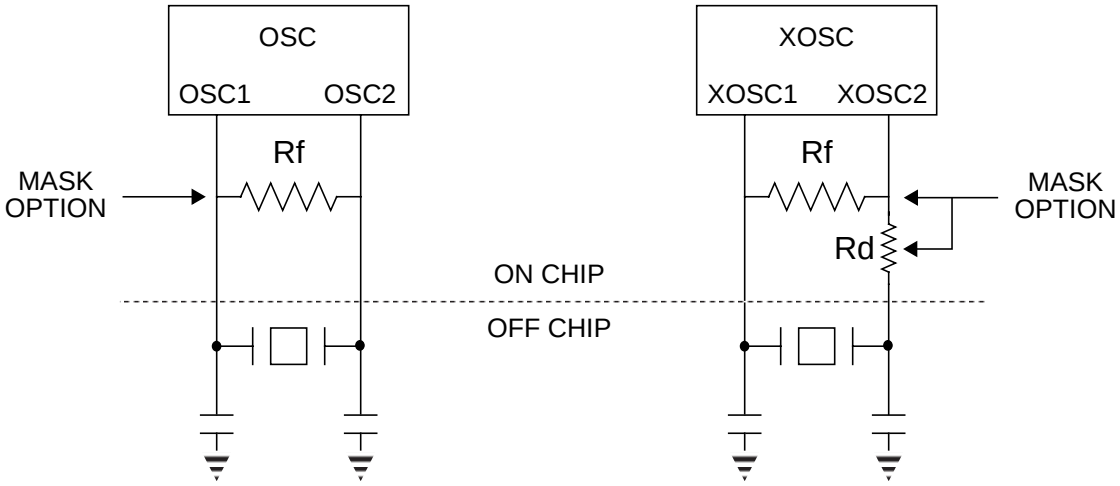


Figure 1-6: OSC1/2 and XOSC1/2 Mask Options

XOSC WITH FOSCE=1

If XOSC is the system clock and FOSCE=1, executing the STOP instruction will halt OSC, put the MCU into a low-power mode and clear the 6-bit POD counter. The 7-bit divider is not initialized. Exiting STOP with external IRQ re-starts the oscillator; however, execution begins immediately using XOSC. When the POD counter overflows, FTUP is set signaling that OSC is stable and OSC can be used as the system clock. The stabilization time will vary between 8064 and 8192 counts.

XOSC WITH FOSCE=0

If XOSC is the system clock, clearing FOSCE will stop OSC and preset the 7-bit divider plus the 6-bit POD counter to $0078. Execution will continue with XOSC, and when FOSCE is set again, OSC will re-start. When the POD counter overflows, FTUP is set signaling that OSC is stable and OSC can be used as the system clock. The stabilization time will be 8072 counts.

XOSC WITH FOSCE=0 AND STOP

If XOSC is the system clock and FOSCE is cleared, further power reduction can be achieved by executing the STOP instruction. In this case, OSC is stopped, the 7-bit divider plus the 6-bit POD counter are preset to $0078 (since FOSCE=0) and execution is halted. Exiting STOP with external IRQ does not re-start the OSC; however, execution begins immediately using XOSC. OSC may be re-started by setting FOSCE, and when the POD counter overflows, FTUP be will set signaling that OSC is stable and can be used as the system clock. The stabilization time will be 8072 counts.

1.4.1.3 OSC CLOCK DIVIDER AND POD COUNTER

The OSC clock is divided by a 7-bit counter which is used for the system clock, time base, and POD counter. Clocks divided by 2, 4, and 64 are available for the system clock selections and a clock divided by 128 is provided for the time base and POD counter.

The POD counter is a 6 bit-clock counter that is driven by the OSC divided by 128. The overflow of this counter is used for setting FTUP bit, release of power-on delay (POD), and resuming operation from stop mode.

The 7-bit divider plus the 6-bit POD counter are initialized to $0078 by the following conditions.

- Power-on detection
- When FOSCE bit is cleared

1.4.1.4 SYSTEM CLOCK CONTROL

The system clock is provided for all internal modules except time base.

Both OSC and XOSC are available as the system clock source. The divide ratio is selected by the SYS1 and SYS0 bits in the MISC register.

By default OSC divided by two is selected on reset.

Table 1-1: System Clock Frequency

SYS1	SYS0	DIVIDE RATIO	FREQUENCY (HZ)		
			OSC= 4.0M	OSC= 4.1943M	XOSC= 32.768K
0	0	OSC DIVIDED BY 2	2.0M	2.0972M	----
0	1	OSC DIVIDED BY 4	1.0M	1.0486M	----
1	0	OSC DIVIDED BY 64	62.5K	65.536K	----
1	1	XOSC DIVIDED BY 2	----	----	16.384K

1.4.1.5 STOP AND WAIT MODES

During stop mode, the main oscillator (OSC) is shut down and the clock pass from the second oscillator (XOSC) is disconnected so that all modules except time base are halted. Entering stop mode clears FTUP flag in the MISC register and initializes the POD counter. The stop mode is exited by the RESET, $\overline{\text{IRQ1/2}}$, KWI, SPI1/2 (slave mode), or TB interrupt (TBCLK=0).

If OSC is selected as the system clock source during stop mode, CPU resumes after the overflow of the POD counter and this overflow sets FTUP status flag.

If XOSC is selected as system clock source during stop mode, no stop recovery time is required for exiting stop mode because XOSC never stops and re-start of main oscillator depends on FOSCE bit.

During wait mode, only the CPU clocks are halted and the peripheral modules are bit affected. The wait mode is exited by RESET or any interrupts.

Table 1-2: Recovery Time Requirements

BEFORE RESET OR INTERRUPT			POWER ON	EXTERNAL RESET	EXIT STOP MODE BY INTERRUPT
CPU CLOCK SOURCE	STOP	FOSCE			
----------	-------	-------	WAIT	----------	----------
OSC (OSC ON)	OUT	1	----------	NO WAIT	----------
OSC (OSC OFF)	OUT IN IN *2	0 *1 1 0 *2	---------- ---------- ----------	WAIT WAIT WAIT	---------- WAIT WAIT
XOSC (OSC ON)	OUT	1	----------	NO WAIT	----------
XOSC (OSC OFF)	OUT IN IN	0 1 0	---------- ---------- ----------	WAIT WAIT WAIT	---------- NO WAIT NO WAIT

*1 THIS CASE HAS NO MEANING FOR THE APPLICATIONS
*2 THIS CASE NEVER OCCURS

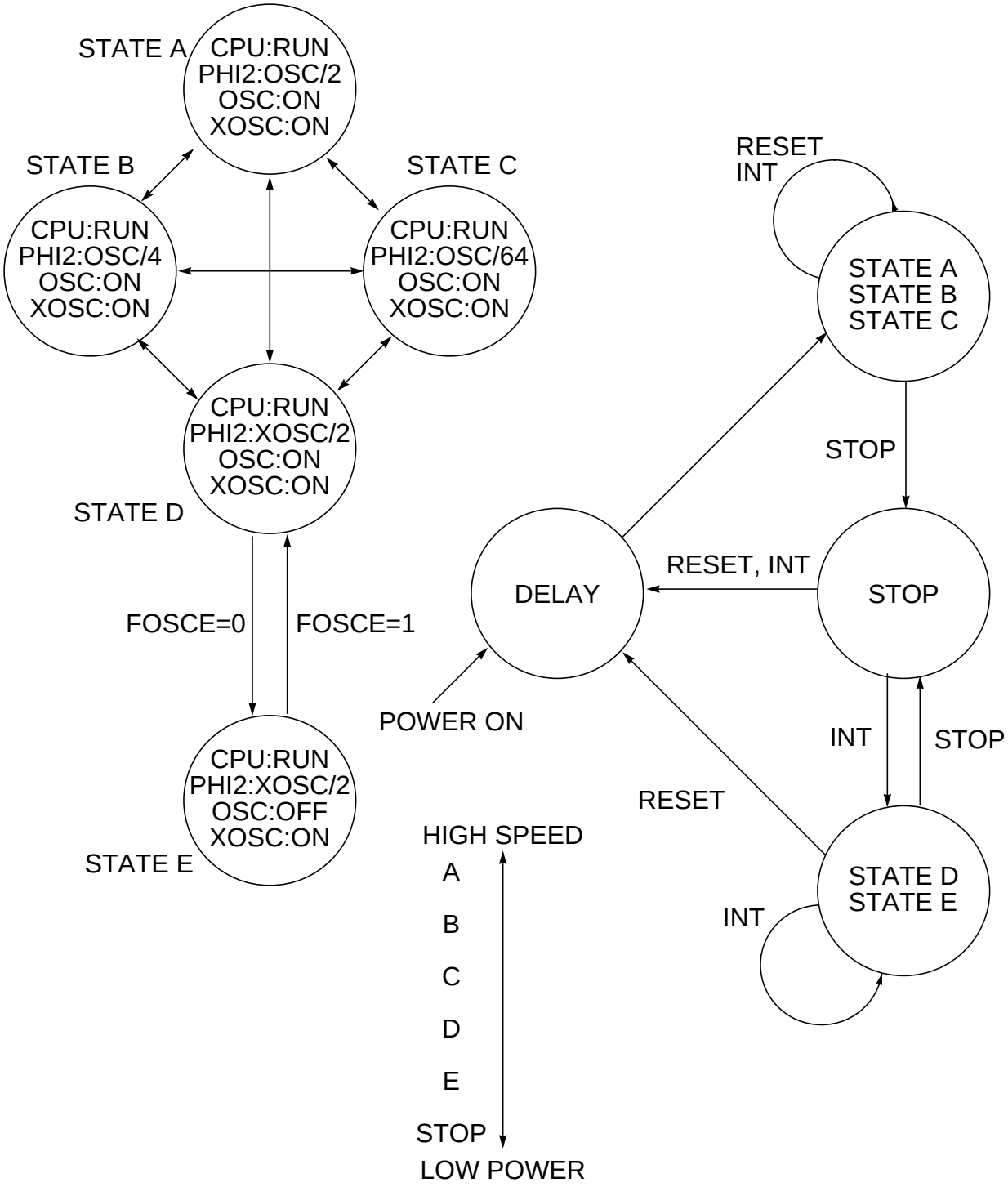


Figure 1-7: Clock State and STOP/POD Delay Diagram

1.4.1.6 TIME BASE

Time base is a 14-bit up counter which is clocked by XOSC input or OSC input divided by 128. The TBCLK bit in the TBCR1 register selects the clock source.

This divider is initialized to $0078 only on power-on delay. After counting 8072 clocks, STUP bit in the MISC register is set.

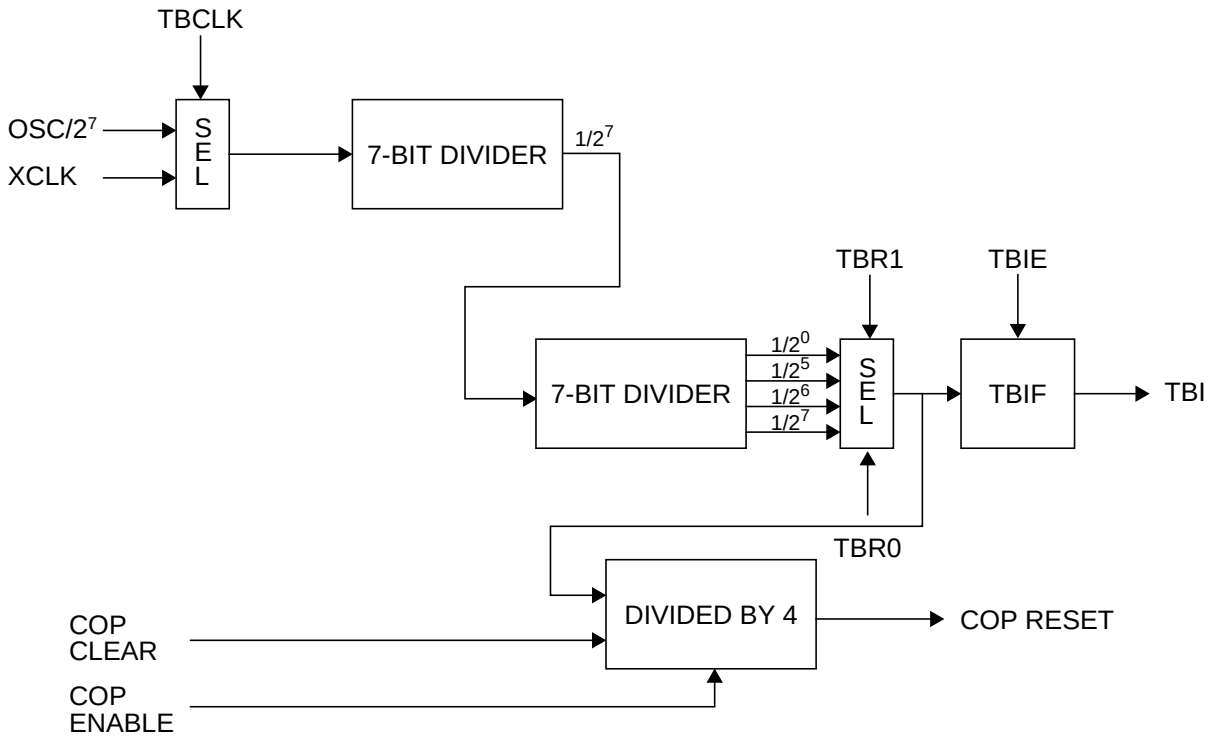


Figure 1-8: Time Base Clock Divider

The divided clocks from the time base are used as follows:

STUP — Time base divider is initialized to $0078 by the power-on detection. When the count reaches 8072, the STUP flag in the MISC register is set. Once STUP flag is set, it is never cleared until power down.

TBI — Time base interrupt may be generated at every 0.5, 0.25, 0.125, or 0.0039 seconds with 32.768 KHz crystal at XOSC pins.

Time base interrupt flag (TBIF) is set at every period and interrupt is requested if the enable bit (TBIE) is set. The clock divided by 128, 4096, 8192, or 16,384 is used to set TBIF, and this clock is selected by the TBR1 and TBRO bits in the TBCR2 register.

Table 1-3: TB Interrupt Frequency

TBR1	TBR0	DIVIDE RATIO	FREQUENCY (HZ)		
			OSC= 4.0M	OSC= 4.1943M	XOSC= 32.768K
0 0 1 1	0 1 0 1	TBCLK DIVIDED BY 128 TBCLK DIVIDED BY 4096 TBCLK DIVIDED BY 8192 TBCLK DIVIDED BY 16,384	244 7.63 3.81 1.91	256 8.00 4.00 2.00	256 8.00 4.00 2.00

COP

The computer operating properly (COP) watchdog timer is controlled by the COPE and COPC bits in the TBCR2 register.

The COP uses the same clock as TBI that is selected by the TBR1 and TBR0 bits. The TBI is divided by four and overflow of this divider generates COP timeout reset if the COP enable (COPE) bit is set. The COP timeout reset has the same vector address as POD and external RESET. To prevent the COP timeout, the COP divider is cleared by writing a one to the COP clear (COPC) bit.

When the time base divider is driven by the OSC clock, the clock for the divider is suspended during stop mode or when FOSCE is 0. This may cause COP period stretching or no COP timeout reset when processing errors occur. To avoid these problems, it is recommended that XOSC clock be used for the COP functions.

When the time base (COP) divider is driven by the XOSC clock, the divider does not stop counting and the COPC bit must be triggered to prevent the COP timeout.

Table 1-4: COP Timeout Period

TBR1	TBR0	COP PERIOD (MILLI-SECOND)					
		OSC=4.0MHz		OSC=4.1943MHz		XOSC=32.768KHz	
		MIN	MAX	MIN	MAX	MIN	MAX
0 0 1 1	0 1 0 1	12.3 393 786 1573	16.4 524 1048 2097	11.7 375 750 1500	15.6 500 1000 2000	11.7 375 750 1500	15.6 500 1000 2000

1.4.1.7 TIME BASE CONTROL REGISTER 1 (TBCR1)

	B7	B6	B5	B4	B3	B2	B1	B0	
$0010	TBCLK	0	0	0	T3R1	T3R0	T2R1	T2R0	TBCR1
RESET:	0	0	0	0	0	0	0	0	

READ: anytime

WRITE: anytime (Only one-time write is allowed on bit 7 after reset.)

TBCLK Time Base Clock

The TBCLK bit selects time base clock source. This bit is cleared at reset. After reset, write to this bit is allowed only once.

0 - XOSC clock is selected
1 - OSC clock divided by 128 is selected

BITS 6-4 Reserved

These bits are not used and always read as zero.

T3R1/0 Prescale Rate or Clock select bits for PWM

These 2 bits select the clock for the PWM. (See **8.5.5 TIMER BASE CONTROL REGISTER 1 (TBCR1).**)

T2R1/0 Preschool Rate Select bits for Timer 2

These 2 bits select the timer 2 clock rate. (See **8.5.5 TIMER BASE CONTROL REGISTER 1 (TBCR1)**.)

1.4.1.8 TIME BASE CONTROL REGISTER 2 (TBCR2)

	B7	B6	B5	B4	B3	B2	B1	B0	
$0011	TBIF	TBIE	TBR1	TBR0	RTBIF	0	COPE	COPC	TBCR2
RESET:	0	0	1	1	0	0	0	0	

READ: anytime (Bits 3 and 0 are write-only bits and always read as zero.)

WRITE: anytime (Bit 7 is a read-only bit and write has no effect; bit 1 is a one-time write bit.)

TBIF **Time Base Interrupt Flag**

The TBIF bit is set every timeout interval of the time base. This is a read-only bit and is cleared by writing a one to the RTBIF bit. Reset clears the TBIF bit. Time base interrupt period between reset and first TBIF depends on the time elapsed during reset, since the time base divider is not initialized by reset.

TBIE **Time Base Interrupt Enable**

The TBIE bit enables the time base interrupt capability. If TBIF = 1 and TBIE = 1, the time base interrupt is generated.

0 - TB interrupt is disabled
1 - TB interrupt requested when TBIF = 1

TBR1/0 **Time Base Interrupt Rate Select**

The TBR1 and TBR0 bits select one of four rates for the time base interrupt period. The TB interrupt rate is also related to the COP reset period. These bits are set to one by reset.

TBR1	TBR0	DIVIDE RATIO	FREQUENCY (HZ)		
			OSC= 4.0M	OSC= 4.1943M	XOSC= 32.768K
0	0	TBCLK DIVIDED BY 128	244	256	256
0	1	TBCLK DIVIDED BY 4096	7.63	8.00	8.00
1	0	TBCLK DIVIDED BY 8192	3.81	4.00	4.00
1	1	TBCLK DIVIDED BY 16,384	1.91	2.00	2.00

RTBIF **Reset TB Interrupt Flag**

The RTBIF bit is a write-only bit and always read as zero. Writing a one to this bit clears the TBIF bit and writing zero to this bit has no effect.

BIT 2 **Reserved**

This bit is not used and always read as zero.

COPE **COP Enable**

When the COPE bit is 1, COP reset function is enabled. This bit is cleared by the reset (including COP reset) and write to this bit is allowed only one time after reset.

COPC **COP Clear**

Writing a one to COPC bit clears the 2-bit divider to prevent COP timeout. (The COP timeout period depends on the TB interrupt rate.) This bit is write-only and returns to zero when read.

1.4.1.9 MISCELLANEOUS REGISTER (MISC)

	B7	B6	B5	B4	B3	B2	B1	B0	
$003E	FTUP	STUP	0	0	SYS1	SYS0	FOSCE	OPTM	MISC
RESET:	*	*	0	0	0	0	1	0	

READ: anytime

WRITE: Bits 7-4: no effect
Bits 3-1: anytime (Software must take care of changing these bits.)
Bit 0: anytime

FTUP OSC Time Up Flag

Power-on detection and clearing FOSCE bit clears this bit. This bit is set by the overflow of the POD counter. The external reset does not affect this bit.

READ**:**

0 - during POD or OSC shut down
1 - OSC clock is available for the system clock

STUP XOSC Time Up Flag

The power-on detection clears this bit. This bit is set after the time base has counted 8072 clocks. The external reset does not affect this bit.

READ**:**

0 - XOSC is not stabilized or no connection on XOSC1/2 pins
1 - XOSC clock is available for the system clock

BITS 5-4 Reserved

These bits are not used and always read as zero.

SYS1/0 System Clock Select

These 2 bits select the system clock source. Upon reset, the SYS1 and SYS0 bits are initialized to zeros.

SYS1	SYS0	DIVIDE RATIO	FREQUENCY (HZ)		
			OSC= 4.0M	OSC= 4.1943M	XOSC= 32.768K
0	0	OSC DIVIDED BY 2	2.0M	2.0972M	----
0	1	OSC DIVIDED BY 4	1.0M	1.0486M	----
1	0	OSC DIVIDED BY 64	62.5K	65.536K	----
1	1	XOSC DIVIDED BY 2	----	----	16.384K

FOSCE Fast (Main) Oscillator Enable

The FOSCE bit controls main oscillator activity. This bit should not be cleared by the CPU when the main oscillator is selected as the system clock source.

When this bit is cleared:

1. OSC is shut down.
2. 7-bit divider at the OSC input plus 6-bit POD counter are initialized to $0078.
3. FTUP flag is cleared.

When this bit is set:

1. Main oscillator starts again.
2. FTUP flag is set by the POD counter overflow (8072 clocks).

OPTM Option Map Select

The OPTM bit selects one of two register maps at $0000-$000F. This bit is cleared on reset.

0 - Main register map is selected
1 - Option map is selected

1.5 TST/V_{PP} PIN

In the normal operation mode (SCM), this pin should be tied to V_{DD} level.

1.5.1 SUMMARY OF INTERNAL REGISTERS AND I/O MAP

The following figure explains how to interpret the register figures used in this document.

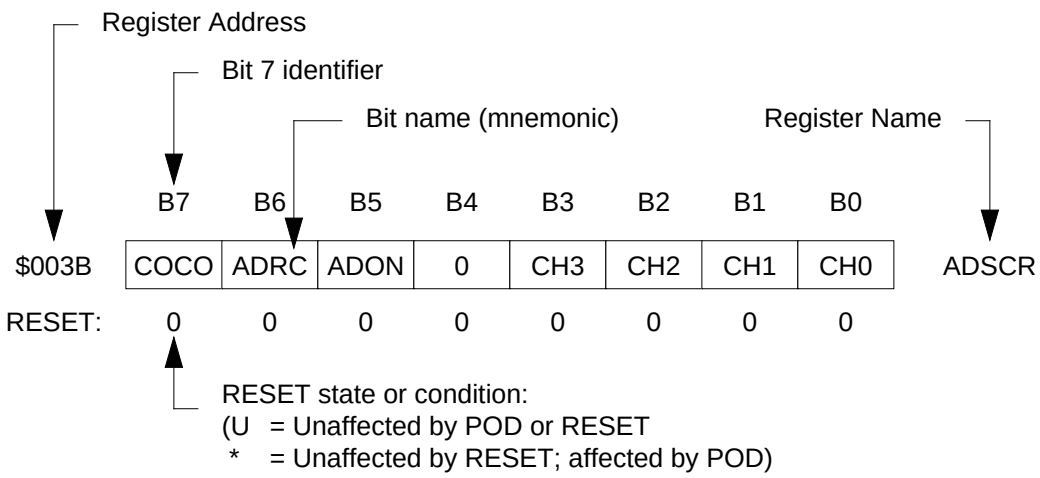


Figure 1-9: Register Description Key

INTERNAL REGISTERS -- MAIN I/O MAP (OPTM = 0)

	B7	B6	B5	B4	B3	B2	B1	B0	
$0000	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0	PORTA
$0001	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0	PORTB
$0002	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0	PORTC
$0003	PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0	PORTD
$0004	PE7	PE6	PE5	PE4	PE3	PE2	PE1	PE0	PORTE
$0005	PF7	PF6	PF5	PF4	PF3	PF2	PF1	PF0	PORTF
$0006	PG7	PG6	PG5	PG4	PG3	PG2	PG1	PG0	PORTG
$0007	PH7	PH6	PH5	PH4	PH3	PH2	PH1	PH0	PORTH
$0008	IRQ1E	IRQ2E	0	KWIE	IRQ1S	IRQ2S	0	0	INTCR
$0009	IRQ1F	IRQ2F	0	KWIF	RIRQ1	RIRQ2	0	RKWIF	INTSR
$000A	SPIE1	SPE1	DORD1	MSTR1	0	0	0	SPR1	SPCR1
$000B	SPIF1	DCOL1	0	0	0	0	0	0	SPSR1
$000C	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0	SPDR1
$000D	SPIE2	SPE2	DORD2	MSTR2	0	0	0	SPR2	SPCR2
$000E	SPIF2	DCOL2	0	0	0	0	0	0	SPSR2
$000F	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0	SPDR2
	B7	B6	B5	B4	B3	B2	B1	B0	

Figure 1-10: Main I/O Map ($0000-$000F)

INTERNAL REGISTERS — I/O MAP

	B7	B6	B5	B4	B3	B2	B1	B0	
$0010	TBCLK	0	0	0	T3R1	T3R0	T2R1	T2R0	TBCR1
$0011	TBIF	TBIE	TBR1	TBR0	RTBIF	0	COPE	COPC	TBCR2
$0012	ICIE	OC1IE	TOIE	0	0	0	IEDG	OLVL	TCR
$0013	ICF	OC1F	TOF	0	0	0	0	0	TSR
$0014	BIT15	BIT14	BIT13	BIT12	BIT11	BIT10	BIT9	BIT8	ICH
$0015	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0	ICL
$0016	BIT15	BIT14	BIT13	BIT12	BIT11	BIT10	BIT9	BIT8	OC1H
$0017	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0	OC1L
$0018	BIT15	BIT14	BIT13	BIT12	BIT11	BIT10	BIT9	BIT8	TCNTH
$0019	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0	TCNTL
$001A	BIT15	BIT14	BIT13	BIT12	BIT11	BIT10	BIT9	BIT8	ACNTH
$001B	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0	ACNTL
$001C	TI2IE	OC2IE	0	T2CLK	IM2	IL2	OE2	OL2	TCR2
$001D	TI2F	OC2F	0	0	RTI2F	ROC2F	0	0	TSR2
$001E	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0	OC2
$001F	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0	TCNT2
	B7	B6	B5	B4	B3	B2	B1	B0	

Figure 1-11: Main I/O Map ($0010-$001F)

NOTE: Main I/O map from $0020-$0033 is reserved for future use.

	B7	B6	B5	B4	B3	B2	B1	B0	
$0034	0	0	0	0	CH3	CH2	CH1	CH0	PWMCR
$0035	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0	PWMCNT
$0036	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0	PWMDR0
$0037	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0	PWMDR1
$0038	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0	PWMDR2
$0039	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0	PWMDR3
$003A	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0	ADR
$003B	COCO	ADRC	ADON	0	CH3	CH2	CH1	CH0	ADSCR
$003C	0	0	0	0	PJ3	PJ2	PJ1	PJ0	PORTJ
$003D	-	-	-	-	-	0	ELAT	PGM	PCR
$003E	FTUP	STUP	0	0	SYS1	SYS0	FOSCE	OPTM	MISC
$003F									(reserved)
	B7	B6	B5	B4	B3	B2	B1	B0	

Figure 1-12: Main I/O Map ($0034-$003F)

1.5.2 OPTION MAP FOR THE I/O CONFIGURATIONS

In MC68HC05G3 (705G4), most of the mask options are replaced by the control register bits to eliminate the problems of emulator, testing, complications of the application support, mask sets, etc. These control bits are implemented in the second register map (option map), which is switched by a register bit.

Some options still remain as mask options such as pileup resistor for $\overline{\text{RESET}}$ pin and resistors for OSC1/2 and XOSC1/2 pins. The status of these mask options can be read using the MOSR in the option map.

The option map is located at $0000-$000F of the main memory map and is available when OPTM bit in the MISC register is set. Main registers at $0000-$000F are not available during OPTM = 1.

Data direction registers are available in the option map.

SYSTEM CONFIGURATION -- OPTION MAP
(OPTM= 1)

	B7	B6	B5	B4	B3	B2	B1	B0	
$0000	DDRA7	DDRA6	DDRA5	DDRA4	DDRA3	DDRA2	DDRA1	DDRA0	DDRA
$0001									(reserved)
$0002	DDRC7	DDRC6	DDRC5	DDRC4	DDRC3	DDRC2	DDRC1	DDRC0	DDRC
$0003	DDRD7	DDRD6	DDRD5	DDRD4	DDRD3	DDRD2	DDRD1	DDRD0	DDRD
$0004	DDRE7	DDRE6	DDRE5	DDRE4	DDRE3	DDRE2	DDRE1	DDRE0	DDRE
$0005									(reserved)
$0006	DDRG7	DDRG6	DDRG5	DDRG4	DDRG3	DDRG2	DDRG1	DDRG0	DDRG
$0007	DDRH7	DDRH6	DDRH5	DDRH4	DDRH3	DDRH2	DDRH1	DDRH0	DDRH
$0008	RHH	RHL	RGH	RGL	RBH	RBL	RAH	RAL	RCR1
$0009	RC7	RC6	RC5	RC4	RC3	RC2	RC1	RC0	RCR2
$000A	0	0	HWOMH	HWOML	GWOMH	GWOML	AWOMH	AWOML	WOM1
$000B	1	1	CWOM5	CWOM4	CWOM3	CWOM2	CWOM1	CWOM0	WOM2
$000C									(reserved)
$000D									(reserved)
$000E	KWIE7	KWIE6	KWIE5	KWIE4	KWIE3	KWIE2	KWIE1	KWIE0	KWIEN
$000F	RSTR	OSCR	XOSCR	0	0	0	0	0	MOSR

Figure 1-13: Option Map ($0000-$000F)

1.5.2.1 RESISTOR CONTROL REGISTER 1 (RCR1)

	B7	B6	B5	B4	B3	B2	B1	B0	
$0008	RHH	RHL	RGH	RGL	RBH	RBL	RAH	RAL	RCR1
RESET:	0	0	0	0	0	0	0	0	

READ: anytime

WRITE: anytime

RHH Port H Pullup Resistor (H)

When this bit is set to one, pullup resistors are connected to the upper four bits of port H pins. This bit is cleared on reset.

RHL Port H Pullup Resistor (L)

When this bit is set to one, pullup resistors are connected to the lower four bits of port H pins. This bit is cleared on reset.

RGH Port G Pullup Resistor (H)

When this bit is set to one, pullup resistors are connected to the upper four bits of port G pins. This bit is cleared on reset.

RGL Port G Pullup Resistor (L)

When this bit is set to one, pullup resistors are connected to the lower four bits of port G pins. This bit is cleared on reset.

RBH Port B Pullup Resistor (H)

When this bit is set to one, pullup resistors are connected to the upper four bits of port B pins. This bit is cleared on reset.

RBL Port B Pullup Resistor (L)

When this bit is set to one, pullup resistors are connected to the lower four bits of port B pins. This bit is cleared on reset.

RAH Port A Pullup Resistor (H)

When this bit is set to one, pullup resistors are connected to the upper four bits of port A pins. This bit is cleared on reset.

RAL Port A Pullup Resistor (L)

When this bit is set to one, pullup resistors are connected to the lower four bits of port A pins. This bit is cleared on reset.

1.5.2.2 RESISTOR CONTROL REGISTER 2 (RCR2)

	B7	B6	B5	B4	B3	B2	B1	B0	
$0009	RC7	RC6	RC5	RC4	RC3	RC2	RC1	RC0	RCR2
RESET:	0	0	0	0	0	0	0	0	

READER: anytime

WRITE: anytime

RC*x* **Port C Pullup Resistor (Bit *x*)**

When RC*x* bit is set to one, the pullup resistor is connected to the corresponding bit of port C pin. This bit is cleared on reset.

1.5.2.3 OPEN DRAIN OUTPUT CONTROL REGISTER 1 (WOM1)

	B7	B6	B5	B4	B3	B2	B1	B0	
$000A	0	0	HWOMH	HWOML	GWOMH	GWOML	AWOMH	AWOML	WOM1
RESET:	0	0	0	0	0	0	0	0	

READ: anytime

WRITE: anytime

BITS 7-6 Reserved

These bits are not used and always return to zero.

HWOMH Port H Open Drain Mode (H)

When this bit is set to one, upper four bits of port H are configured as open drain outputs if corresponding DDRH bit is set to one. This bit is cleared on reset.

HWOML Port H Open Drain Mode (L)

When this bit is set to one, the lower four bits of port H are configured as open drain outputs if the corresponding DDRH bit is set to one. This bit is cleared on reset.

GWOMH Port G Open Drain Mode (H)

When this bit is set to one, the upper four bits of port G are configured as open drain outputs if the corresponding DDRG bit is set to one. This bit is cleared on reset.

GWOML Port G Open Drain Mode (L)

When this bit is set to one, the lower four bits of port G are configured as open drain outputs if the corresponding DDRG bit is set to one. This bit is cleared on reset.

AWOMH Port A Open Drain Mode (H)

When this bit is set to one, the upper four bits of port A are configured as open drain outputs if the corresponding DDRA bit is set to one. This bit is cleared on reset.

AWOML Port E Open Drain Mode (L)

When this bit is set to one, the lower four bits of port A are configured as open drain outputs if the corresponding DDRA bit is set to one. This bit is cleared on reset.

1.5.2.4 OPEN DRAIN OUTPUT CONTROL REGISTER 2 (WOM2)

	B7	B6	B5	B4	B3	B2	B1	B0	
$000B	1	1	CWOM5	CWOM4	CWOM3	CWOM2	CWOM1	CWOM0	WOM2
RESET:	1	1	0	0	0	0	0	0	

READ: anytime

WRITE: anytime

BITS 7-6 Port C Open Drain Mode (Bits 7-6)

These bits are fixed to one, so PC7-6 are always open drain outputs if DDRC7-6 is set to one. These bits are not affected by reset.

CWOM*x* Port C Open Drain Mode (Bit *x*)

When CWOM*x* bit is set to one, port C*x* is configured as an open drain output if DDRC*x* is set to one. This bit is cleared on reset.

1.5.2.5 KEY WAKEUP INPUT ENABLE REGISTER (KWIE)

	B7	B6	B5	B4	B3	B2	B1	B0	
$000E	KWIE7	KWIE6	KWIE5	KWIE4	KWIE3	KWIE2	KWIE1	KWIE0	KWIEN
RESET:	0	0	0	0	0	0	0	0	

READ: anytime

WRITE: anytime

KWIE*x* **Key Wakeup Input Enable (Bit *x*)**

When KWIE*x* bit is set to one, KWI*x* (PB*x*) input is enabled for key wakeup interrupt. This bit is cleared on reset.MASK OPTION STATUS REGISTER (MOSR)

	B7	B6	B5	B4	B3	B2	B1	B0	
$000F	RSTR	OSCR	XOSCR	0	0	0	0	0	MOSR
RESET:	U	U	U	0	0	0	0	0	

READ: anytime

WRITE: no effect

RSTR **$\overline{\text{RESET}}$ Pin Pullup Resistor**

When this bit is set to one, it indicates the pullup resistor is attached to the RESET pin.

OSCR **OSC Feedback Resistor**

When this bit is set to one, it indicates that the feedback resistor is attached between OSC1 and OSC2.

XOSCR **OSC Feedback Resistor**

When this bit is set to one, it indicates that the feedback resistor is attached between XOSC1 and XOSC2, and the damping resistor at the XOSC2 pin is attached.

BITS 4-0 **Reserved**

These bits are not used and always return to zero.

SECTION 2 MODES OF OPERATION

2.1 GENERAL

The MC68HC05G3 has two operating modes: single-chip mode and self-check mode. In the MC68HC705G4, the self-check mode becomes bootstrap mode.

The single-chip mode allows maximum use of pins for on-chip peripheral functions.

The self-check capability of MC68HC05G3 provides an internal check to determine if the device is functional.

The bootstrap mode is provided for EPROM programming, dumping EPROM contents, reading programs into the internal RAM, and executing it. This is a very versatile mode because the special purpose program that is bootloaded into the internal RAM essentially has no limitations.

2.2 MODE ENTRY

The mode entry is done at the rising edge of the $\overline{\text{RESET}}$ pin. Once the device enters one of the three modes, the mode only can be changed by external reset not by software.

At the rising edge of the $\overline{\text{RESET}}$ pin, the device latches the states of $\overline{\text{IRQ1}}$ and $\overline{\text{IRQ2}}$ pins and places itself in the specified mode. While the $\overline{\text{RESET}}$ pin is low, all pins are configured as single-chip mode. The following table shows the states of $\overline{\text{IRQ1}}$ and $\overline{\text{IRQ2}}$ pins for each mode.

Table 2-1: Mode Select Summary

MODE	$\overline{\text{RESET}}$	$\overline{\text{IRQ1}}$	$\overline{\text{IRQ2}}$
SINGLE-CHIP MODE	↑	L or H	X
SELF-CHECK/BOOTSTRAP		V_{TST}	H

H = V_{DD}
L = GND

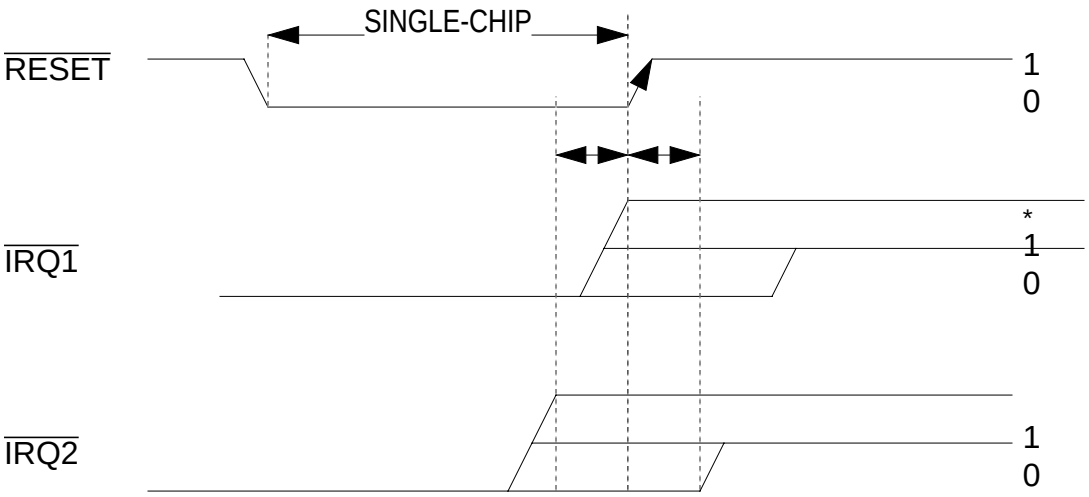


Figure 2-1: HC05G3 (705G4) Mode Entry Diagram

2.3 SINGLE-CHIP MODE (SCM)

In this mode, all address and data bus activity occurs within the MCU so no external pins are required for these functions. The single-chip mode allows the maximum number of I/O pins for on-chip peripheral functions: port A through port J.

2.4 SELF-CHECK/BOOTSTRAP MODE

In this mode, the reset vector is fetched from a 496-byte internal self-check ROM or bootstrap ROM for MC68HC(7)05G4 at $FE00-$FFEF. The self-check ROM contains a self-check program to test the functions of internal modules. The bootstrap ROM contains a small program which reads a program into the internal RAM and then passes control to that program at location $0040, or executes EPROM programming sequence, or dumps EPROM contents.

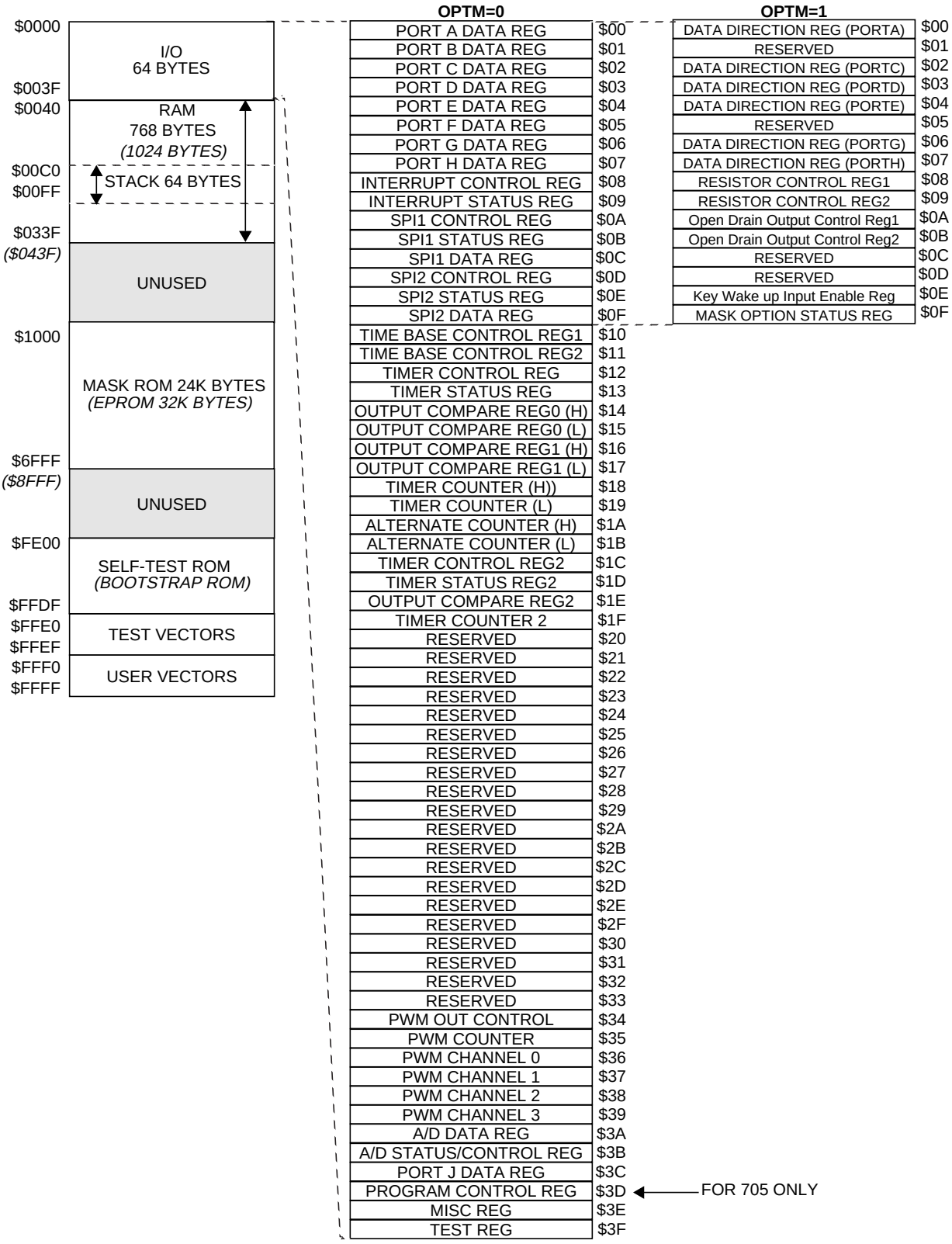
SECTION 3 MEMORY

3.1 GENERAL

The MC68HC05G3 contains 24K mask ROM, 496 bytes of self-check ROM, and 768 bytes of RAM. An additional 16 bytes of mask ROM are provided for user vectors at $FFF0 through $FFFF.

The MC68HC705G4 (EPROM device), contains 32K EPROM, 496 bytes of bootstrap ROM, and 1024 bytes of RAM. An additional 16 bytes of EPROM are provided for user vectors at $FFF0 through $FFFF.

A second set of register map (option map) shares the same memory locations from $0000 to $000F with the main memory map and is available only when the OPTM bit in the MISC register is set. The main memory map at $0000-$000F is not available when OPTM=1. The option map includes the mask option control registers and the data direction registers.



NOTE: Exceptions for the HC705G4 (EPROM device) are in *italic*.

Figure 3-1: MC68HC05G3 (705G4) Memory Map

3.2 RAM

The 768-byte internal RAM is positioned at $0040 through $033F in the memory map. (The 1024-byte internal RAM for MC68HC705G4 is positioned at $0040 through $043F.) The first 192 bytes of memory positioned in page zero are accessible by the direct addressing mode, but the upper 64 bytes of page zero are used for the CPU stack area. Extreme caution should be taken if the stack area is used for data storage.

The RAM is implemented with static cells and retains its contents during the stop and wait modes.

3.3 SELF-CHECK ROM (MC68HC05G3)

Self-check ROM is the 496 bytes of mask ROM positioned at $FE00 through $FFEF. This ROM contains self-check programs and reset/interrupt vectors in the self-check mode.

3.4 BOOT ROM (MC68HC705G4)

Boot ROM is the 496 bytes of mask ROM positioned at $FE00 through $FFEF. This ROM contains bootstrap programs and reset/ interrupt vectors in the bootstrap mode. The programs include:

- EPROM programming and verify
- Dumping EPROM contents
- Reading program into the internal RAM
- Executing program in the internal RAM

3.5 MASK ROM (MC68HC05G3)

The 24K-byte user ROM is positioned at $1000 through $6FFF, and additional 16-byte ROM is located at $FFF0 through $FFFF for user vectors. In this mask ROM device, the V_{PP} pin is not used and the program control register (PCR) is not implemented.

3.6 EPROM (MC68HC705G4)

The 32K-byte EPROM is positioned at $1000 through $8FFF, and additional 16-byte EPROM is located at $FFF0 through $FFFF for user vectors. The erased state of EPROM is read as $FF, and EPROM power is supplied from V_{PP} and V_{DD} pins.

The program control register (PCR) is provided for EPROM programming. EPROM functions are dependent on the device mode.

In user mode, ELAT and PGM bits in the PCR are available for the user programming. The V_{PP} pin should be tied to 5 V or programming voltage.

In the bootstrap mode, all bits of the PCR register are available for the purpose of EPROM programming. The V_{PP} pin should be tied to 5 V or programming voltage.

3.7 PROGRAMMING SEQUENCE

Programming the EPROM of the MC68HC705G4 is similar to programming the MC68HC11A8 EEPROM. The sequence includes:

- Setting the ELAT bit
- Writing the data to the address to be programmed
- Setting the PGM bit
- Delaying for an appropriate amount of time
- Clearing the PGM and the ELAT bit

The last item may be done on a single CPU write. It is important to remember that an external programming voltage must be applied to the V_{PP} pin while programming, but it should be equal to V_{DD} during normal operations.

3.7.1 PROGRAM CONTROL REGISTER (PCR)

Program control register is provided for EPROM programming in the boot modes. This register is available only in the MC68HC705G4 (EPROM device).

	B7	B6	B5	B4	B3	B2	B1	B0	
$003D	-	-	-	-	-	-	ELAT	PGM	PCR
RESET:	0	0	0	0	0	0	0	0	

READ: In user mode, bit 2 through bi 7 read as zero.

WRITE: bit 2 through bit 7 allowed only when in boot mode

BITS 7-2 Reserved

These bits are reserved for factory testing.

ELAT EPROM LATch control

0 - EPROM address and data bus configured for normal reads
1 - EPROM address and data bus configured for programming. (Writes to EPROM cause address and data to be latched. Writes to other areas will not cause any latching.) EPROM is in programming mode and cannot be read if ELAT is 1. This bit may not be set when no V_{PP} voltage is applied to the V_{PP} pin.

PGM EPROM Program Command

0 - Programming power is switched OFF to EPROM array.
1 - Programming power is switched ON to EPROM array.

SECTION 4 CPU CORE

4.1 REGISTERS

The MCU contains five registers as shown in **Figure 4-1: Programming Model**.

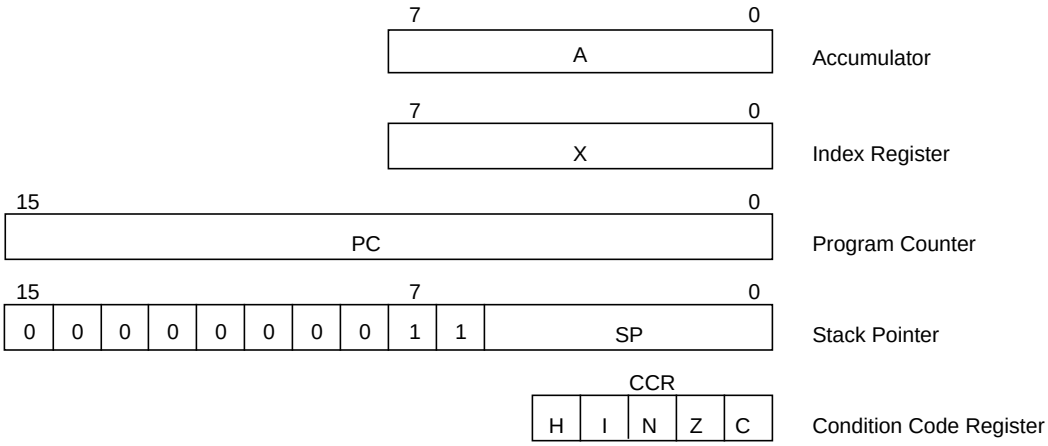


Figure 4-1: Programming Model

4.1.1 ACCUMULATOR (A)

The accumulator is a general-purpose 8-bit register used to hold operands and results of arithmetic calculations or data manipulations.

4.1.2 INDEX REGISTER (X)

The index register is an 8-bit register used for the indexed addressing value to create an effective address. The index register also may be used as a temporary storage area.

4.1.3 PROGRAM COUNTER (PC)

The program counter is a 16-bit register that contains the address of the next byte to be fetched.

4.1.4 STACK POINTER (SP)

The stack pointer contains the address of the next free location on the stack. During an MCU reset or the reset stack pointer (RSP) instruction, the stack pointer is set to location $00FF. The stack pointer then is decremented as data is pushed onto the stack and incremented as data is pulled from the stack.

When accessing memory, the 10 most significant bits are permanently set to 0000000011. These 10 bits are appended to the six least significant bits to produce an address within the range of $00FF to $00C0. Subroutines and interrupts may use up to 64 (decimal) locations. If 64 locations are exceeded, the stack pointer wraps around and loses the previously stored information. A subroutine call occupies two locations on the stack; an interrupt uses five locations. See **Figure 4-2: Stacking Order.**

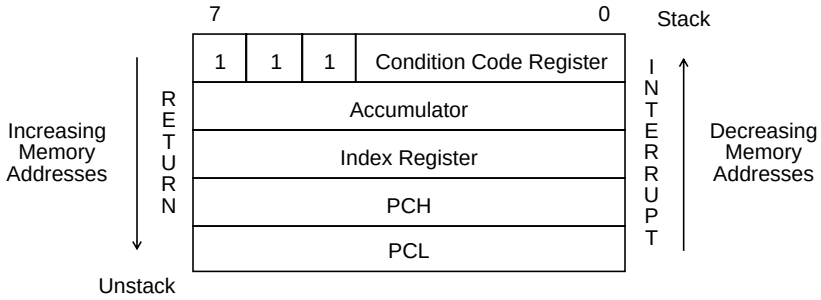


Figure 4-2: Stacking Order

NOTE: Since the stack pointer decrements during pushes, the PCL is stacked first, followed by PCH, etc. Pulling from the stack is in the reverse order.

4.1.5 CONDITION CODE REGISTER (CCR)

The CCR is a 5-bit register in which four bits are used to indicate the results of the instruction just executed, and the fifth bit indicates whether interrupts are masked. These bits can be tested individually by a program, and specific actions can be taken as a result of their state. Each bit is explained in the following paragraphs.

4.1.6 HALF CARRY (H)

This bit is set during ADD and ADC operations to indicate that a carry occurred between bits 3 and 4.

4.1.6.1 INTERRUPT (I)

When this bit is set, the timer and external interrupt are masked (disabled). If an interrupt occurs while this bit is set, the interrupt is latched and processed as soon as the interrupt bit is cleared.

4.1.6.2 NEGATIVE (N)

When set, this bit indicates that the result of the last arithmetic, logical, or data manipulation was negative.

4.1.6.3 ZERO (Z)

When set, this bit indicates that the result of the last arithmetic, logical, or data manipulation was zero.

4.1.6.4 CARRY/BORROW (C)

When set, this bit indicates that a carry or borrow out of the arithmetic logical unit (ALU) occurred during the last arithmetic operation. This bit also is affected during bit test and branch instructions and during shifts and rotates.

4.2 INSTRUCTION SET

The MCU has a set of 62 basic instructions. They can be divided into five different types: register/memory, read-modify-write, branch, bit manipulation, and control. The following paragraphs briefly explain each type. For more information on the instruction set, refer to the *M6805 Family User's Manual* (M6805UM/AD2) or the *MC68HC05C4 Data Sheet* (MC68HC05C4/D).

4.2.1 REGISTER/MEMORY INSTRUCTIONS

Most of these instructions use two operands. One operand is either the accumulator or the index register. The other operand is obtained from memory using one of the addressing modes. The jump unconditional (JMP) and jump to subroutine (JSR) instructions have no register operand. Refer to the following instruction list.

Function	Mnemonic
Load A from Memory	LDA
Load X from Memory	LDX
Store A in Memory	STA
Store X in Memory	STX
Add Memory to A	ADD
Add Memory and Carry to A	ADC
Subtract Memory	SUB
Subtract Memory from A with Borrow	SBC
AND Memory to A	AND
OR Memory with A	ORA
Exclusive OR Memory with A	EOR
Arithmetic compare A with Memory	CMP
Arithmetic Compare X with Memory	CPX
Bit Test Memory with A (Logical Compare)	BIT
Jump Unconditional	JMP
Jump to Subroutine	JSR
Multiply	MUL

4.2.2 READ-MODIFY-WRITE INSTRUCTIONS

These instructions read a memory location or a register, modify or test its contents, and write the modified value back to memory or to the register. The test for negative or zero (TST) instruction is an exception to the read-modify-write sequence since it does not modify the value. Refer to the following list of instructions.

Function	Mnemonic
Increment	INC
Decrement	DEC
Clear	CLR
Complement	COM
Negate (Two's Complement)	NEG
Rotate Left Through Carry	ROL
Rotate Right Through Carry	ROR
Logical Shift Left	LSL
Logical Shift Right	LSR
Arithmetic Shift Right	ASR
Test for Negative or Zero	TST

4.2.3 BRANCH INSTRUCTIONS

This set of instructions branches if a particular condition is met; otherwise, no operation is performed. Branch instructions are 2-byte instructions. Refer to the following list for branch instructions.

Function	Mnemonic
Branch Always	BRA
Branch Never	BRN
Branch if Higher	BHI
Branch if Lower or Same	BLS
Branch if Carry Clear	BCC
Branch if Higher or Same	BHS
Branch if Carry Set	BCS
Branch if Lower	BLO
Branch if Not Equal	BNE
Branch if Equal	BEQ
Branch if Half Carry Clear	BHCC
Branch if Half Carry Set	BHCS
Branch if Plus	BPL
Branch if Minus	BMI
Branch if Interrupt Mask Bit is Clear	BMC
Branch if Interrupt Mask Bit is Set	BMS
Branch if Interrupt Line is Low	BIL
Branch if Interrupt Line is High	BIH
Branch to Subroutine	BSR

4.2.4 BIT MANIPULATION INSTRUCTIONS

The MCU is capable of setting or clearing any writable bit which resides in the first 256 bytes of the memory space where all port registers, port DDRs, timer, timer control, and on-chip RAM reside. An additional feature allows the software to test and branch on the state of any bit within these 256 locations. The bit set, bit clear and bit test, and branch functions are all implemented with a single instruction. For test and branch instructions, the value of the bit tested also is placed in the carry bit of the condition code register. These instructions also are read-modify-write instructions. Do not bit manipulate write-only locations. Refer to the following list for bit manipulation instructions.

Function	Mnemonic
Branch if Bit n is Set	BRSET n (n = 0. . .7)
Branch if Bit n is Clear	BRCLR n (n = 0. . .7)
Set Bit n	BSET n (n = 0. . .7)
Clear Bit n	BCLR n (n = 0. . .7)

4.2.5 CONTROL INSTRUCTIONS

These instructions are register reference instructions and are used to control processor operation during program execution. Refer to the following list for control instructions.

Function	Mnemonic
Transfer A to X	TAX
Transfer X to A	TXA
Set Carry Bit	SEC
Clear Carry Bit	CLC
Set Interrupt Mask Bit	SEI
Clear Interrupt Mask Bit	CLI
Software Interrupt	SWI
Return from Subroutine	RTS
Return from Interrupt	RTI
Reset Stack Pointer	RSP
No-Operation	NOP
Stop	STOP
Wait	WAIT

4.3 ADDRESSING MODES

The MCU uses 10 different addressing modes to provide the programmer with an opportunity to optimize the code for all situations. The various indexed addressing modes make it possible to locate data tables, code conversion tables, and scaling tables anywhere in the memory space. Short indexed accesses are single-byte instructions; the longest instructions (three bytes) permit accessing tables throughout memory. Short and long absolute addressing is also included. One- or 2-byte direct addressing instructions access all data bytes in most applications. Extended addressing permits jump instructions to reach all memory.

The term effective address (EA) is used in describing the various addressing modes. Effective address is defined as the address from which the argument for an instruction is fetched or stored.

4.3.1 IMMEDIATE

In the immediate addressing mode, the operand is contained in the byte immediately following the opcode. The immediate addressing mode is used to access constants that do not change during program execution (for example, a constant used to initialize a loop counter).

4.3.2 DIRECT

In the direct addressing mode, the effective address of the argument is contained in a single byte following the opcode byte. Direct addressing allows the user to directly address the lowest 256 bytes in memory ($0000-$00FF) with a single 2-byte instruction.

4.3.3 EXTENDED

In the extended addressing mode, the effective address of the argument is contained in the two bytes following the opcode byte. Instructions with extended addressing mode are capable of referencing arguments anywhere in memory with a single 3-byte instruction. When using the Motorola assembler, the user need not specify whether an instruction uses direct or extended addressing. The assembler automatically selects the shortest form of the instruction.

4.3.4 RELATIVE

The relative addressing mode is only used in branch instructions. In relative addressing, the contents of the 8-bit signed offset byte, which is the last byte of the instruction, is added to the PC if, and only if, the branch conditions are true. Otherwise, control proceeds to the next instruction. The span of relative addressing is from -127 to +128 from the address of the next opcode. The programmer need not calculate the offset when using the Motorola assembler, since it calculates the proper offset and checks to see that it is within the span of the branch.

4.3.5 INDEXED, NO OFFSET

In the indexed, no offset addressing mode, the effective address of the argument is contained in the 8-bit index register. This addressing mode can access the first 256 memory locations ($0000-$00FF). These instructions are only one byte long. This mode is often used to move a pointer through a table or to hold the address of a frequently referenced RAM or I/O location.

4.3.6 INDEXED, 8-BIT OFFSET

In the indexed, 8-bit offset addressing mode, the effective address is the sum of the contents of the unsigned 8-bit index register and the unsigned byte following the opcode. The addressing mode is useful for selecting the K^{th} element in an n element table. With this 2-byte instruction, K typically would be in X with the address of the beginning of the table in the instruction. As such, tables may begin anywhere within the first 256 addressable locations and could extend as far as location 510 ($01FE). This is the last location which can be accessed in this way.

4.3.7 INDEXED, 16-BIT OFFSET

In the indexed, 16-bit offset addressing mode, the effective address is the sum of the contents of the unsigned 8-bit index register and the two unsigned bytes following the opcode. This address mode can be used in a manner similar to indexed, 8-bit offset except that this 3-byte instruction allows tables to be anywhere in memory. As with direct and extended addressing, the Motorola assembler determines the shortest form of indexed addressing.

4.3.8 BIT SET/CLEAR

In the bit set/clear addressing mode, the bit to be set or cleared is part of the opcode, and the byte following the opcode specifies the direct addressing of the byte in which the specified bit is to be set or cleared. Any read/write bit in the first 256 locations of memory, including I/O, can be selectively set or cleared with a single 2-byte instruction.

4.3.9 BIT TEST AND BRANCH

The bit test and branch addressing mode is a combination of direct addressing and relative addressing. The bit that is to be tested and its condition (set or clear) is included in the opcode. The address of the byte to be tested is in the single byte immediately following the opcode byte. The signed relative 8-bit offset in the third byte is added to the PC if the specified bit is set or cleared in the specified memory location. This single 3-byte instruction allows the program to branch based on the condition of any readable bit in the first 256 locations of memory. The span of branching is from -128 to +127 from the address of the next opcode. The state of the tested bit also is transferred to the carry bit of the condition code register.

4.3.10 INHERENT

In the inherent addressing mode, all the information necessary to execute the instruction is contained in the opcode. Operations specifying only the index register and/or accumulator as well as the control instruction with no other arguments are included in this mode. These instructions are one byte long.

4.4 LOW-POWER MODES

The MC68HC05G3 (705G4) has two power-saving modes, stop and wait. Flowcharts of these modes are shown in **Figure 4-3: STOP/WAIT Flowcharts**.

4.4.1 STOP MODE

The STOP instruction places the MCU in its lowest power consumption mode. During stop mode, the internal main oscillator (OSC) is turned off and the clock pass from the second oscillator (XOSC) is disconnected, so that all modules except time base are halted. The sub-oscillator (XOSC) does not stop oscillating. Therefore, if XOSC is used as the clock source for COP, COP still is functional in stop mode.

During stop mode, the I bit in the CCR is cleared to enable external interrupts. All other registers and memory remain unaltered. All input/output lines remain unchanged. The stop mode is exited by RESET or by receipt of an interrupt from $\overline{\text{IRQ}}$1/2, KWI, SPI1/2 (slave mode only), or TBI (when XOSC is selected as time base clock). Refer to **1.4.1 OSCILLATORS AND CLOCK DISTRIBUTIONS** for more information during stop mode.

4.4.2 WAIT MODE

The WAIT instruction places the MCU in a low-power consumption mode, but the wait mode consumes more power than the stop mode. In the wait mode, only the CPU clocks are halted and it never affects the peripheral modules. During the wait mode, the I bit in the CCR is cleared to enable interrupts. All other registers, memory, and input/output lines remain in their previous state. The wait mode is exited by RESET and any interrupts. The on-chip oscillator (OSC and XOSC) circuit remains active throughout the wait standby period.

The reduction of power in the wait mode depends on how many of the on-chip peripheral functions can be shut down (clocks). The CPU always shuts down in the wait mode. The peripherals are enabled or disabled based upon their control bits. (The time base clock dividers are always enabled.)

It should be obvious that the amount of power that will be consumed is dependant on the particular application and that it would be prohibitive to test all parts for all variations. For these reasons, the data sheet will include values for a limited number of variations. These variations and the corresponding MAX power consumptions will be decided after the initial characterization of the silicon.

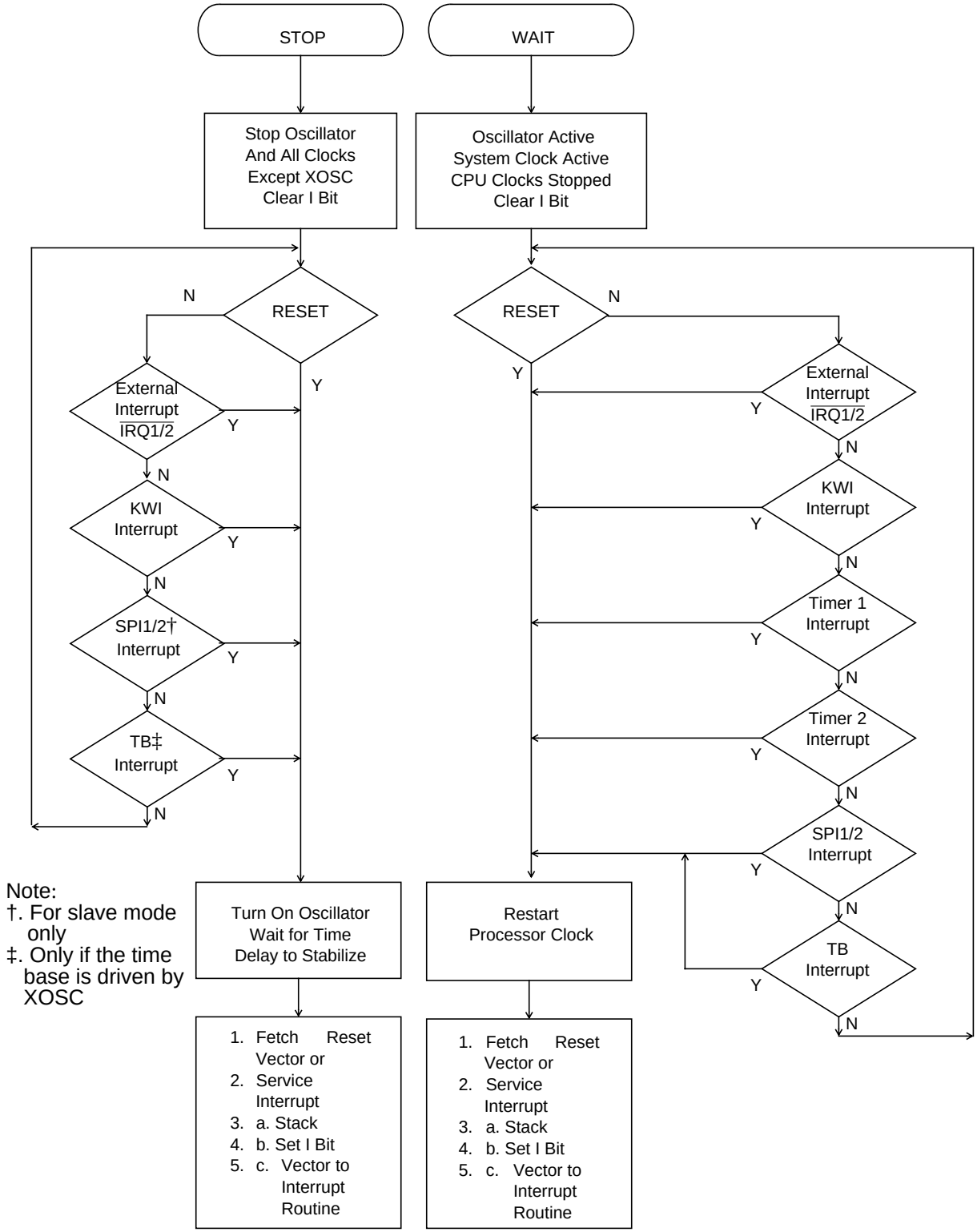


Figure 4-3: STOP/WAIT Flowcharts

THIS PAGE INTENTIONALLY LEFT BLANK

SECTION 5 RESET/ INTERRUPT STRUCTURE

5.1 GENERAL

In user operating modes, the reset/interrupt vectors are located at the top of the address space ($FFF0 through $FFFF). In the self-check (bootstrap) mode, the reset/interrupt vectors are located at $FFE0 through $FFEF in the internal self-check (bootstrap) ROM. For the remainder of this section, a user operating mode will be assumed. The following table shows the address assignments for the vectors.

Table 5-1: Interrupt Vector Assignments

VECTOR ADDRESS	INTERRUPT SOURCE		MASKED BY	LOCAL MASK	PRIORITY (1 = HIGHEST)
FFF0-F1	TBI		I BIT	TBIE	7
FFF2-F3	SPI	SPI1	I BIT	SPIE1	6
		SPI2	I BIT	SPIE2	6
FFF4-F5	TIMER 2	TI2I	I BIT	TI2IE	5
		OC2I	I BIT	OC2IE	5
FFF6-F7	TIMER 1	ICI	I BIT	ICIE	4
		OC1I	I BIT	OC1IE	4
		TOI	I BIT	TOIE	4
FFF8-F9	KWI		I BIT	KWIE	3
FFFA-FB	IRQ	IRQ1	I BIT	IRQ1E	2
		IRQ2	I BIT	IRQ2E	2
FFFC-FD	SWI		NONE	NONE	*
FFFE-FF	RESET	COP	NONE	COPE	1
		RESET PIN	NONE	NONE	1

* Same level as an instruction

Upon reset, the I bit in the condition code register is set and no interrupts are recognized. Also, when an interrupt occurs, the I bit automatically is set by hardware after stacking the CC byte. All interrupts in the MC68HC05G3 (705G4) follow a fixed hardware priority circuit to resolve simultaneous requests. Each of these sources is an input to the priority resolution circuit.

Each interrupt has a software programmable interrupt mask bit which may be used to selectively inhibit automatic hardware response. In addition, the I bits in the condition code register act as class inhibit masks to inhibit all sources in the I bit class. The RESET and software interrupt (SWI) are not masked by the I bit in the condition code register.

5.1.1 SOFTWARE INTERRUPT (SWI)

SWI is an executable instruction rather than a prioritized asynchronous interrupt source. In a sense, it is lower in priority than any source because once any interrupt sequence has begun, SWI cannot override it. In another sense, it is higher in priority than any sources except reset because once the SWI opcode is fetched, no other sources can be honored until after the first instruction in the SWI service routine has been executed. The interrupt service routine address is specified by the contents of memory locations $FFFC and $FFFD. SWI causes the I mask bit in the CC register to be set.

5.2 INTERRUPTS OF THE MC68HC05G3 (705G4)

The HC05G3 (705G4) has six hardware interrupt sources: IRQ1 and IRQ2, key wakeup interrupt (KWI), timer 1 (TOI, ICI, OC1I), timer 2 (TI2I, OC2I), serial transfer complete interrupt (SPI1 and SPI2), and time base interrupt (TBI).

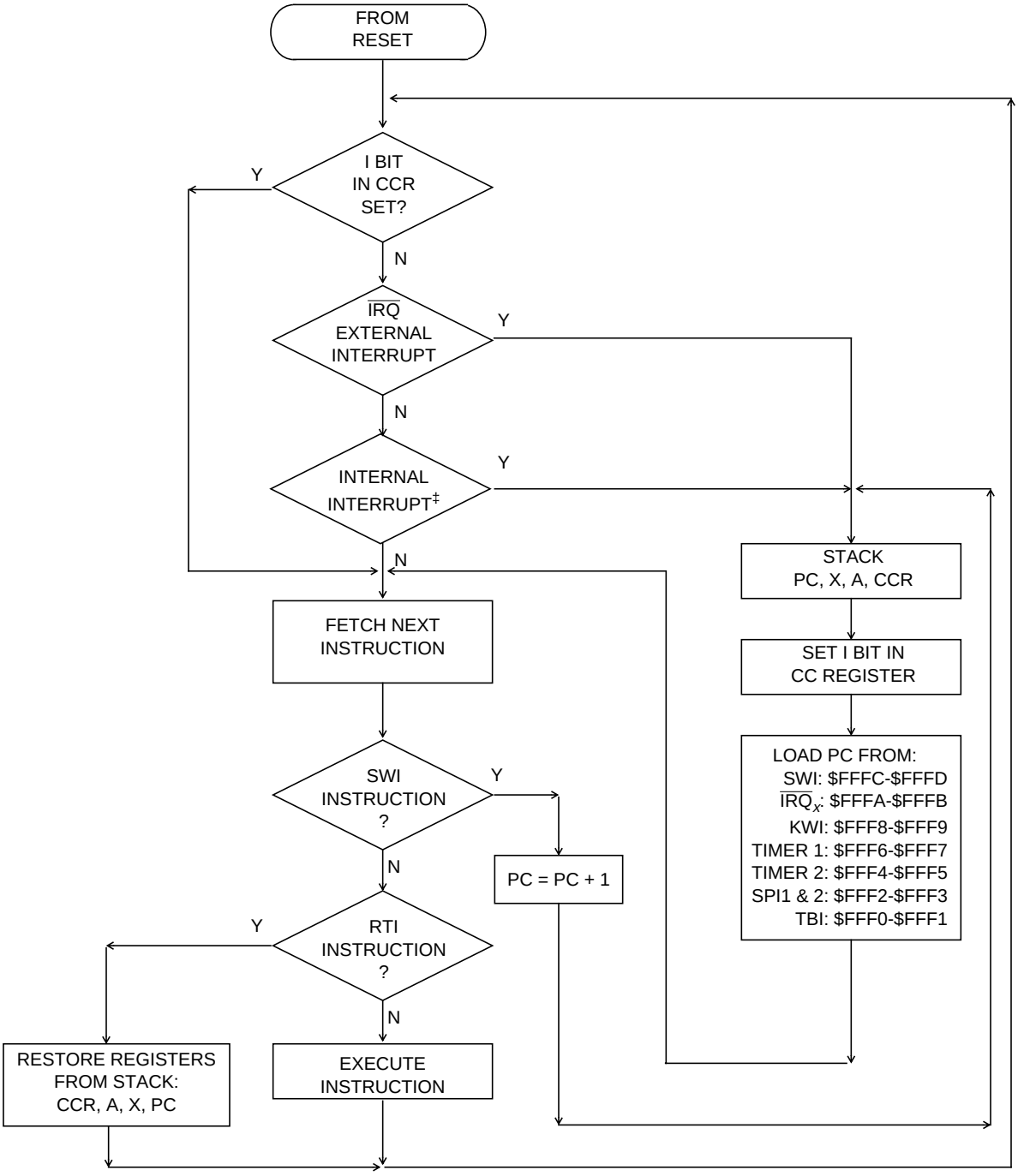
5.2.1 $\overline{\text{IRQ1}}$/$\overline{\text{IRQ2}}$

The two interrupt request inputs, $\overline{\text{IRQ1}}$ and $\overline{\text{IRQ2}}$, share same the vector address at $FFFA and $FFFB.

Two IRQ1S and IRQ2S bits in the interrupt control register (INTCR) control two IRQs, respectively, so that $\overline{\text{IRQ1}}$ and $\overline{\text{IRQ2}}$ respond only to the falling edge or falling edge and low level at the pin. $\overline{\text{IRQ1}}$ and $\overline{\text{IRQ2}}$ are enabled by IRQ1E and IRQ2E bits and IRQ1F and IRQ2F bits are provided in the interrupt status register (INTSR).

$\overline{\text{IRQ1}}$ and $\overline{\text{IRQ2}}$ pins are shared with PC7 and PC6 and the IRQ*x* pin states can be determined by reading port C pins when DDRC7/6 = 0. The BIL and BIH instructions are only effective for the $\overline{\text{IRQ1}}$ input.

When DDRC7/6 =1, the IRQ*x*F can be set by the data latch. Therefore, care must be taken to ensure the flag is cleared by software before the IRQ*x*E bit is enabled.



Note:
‡ KWI, Timer 1 and 2, SPI1 and 2, and TBI

Figure 5-1: Interrupt Flowchart

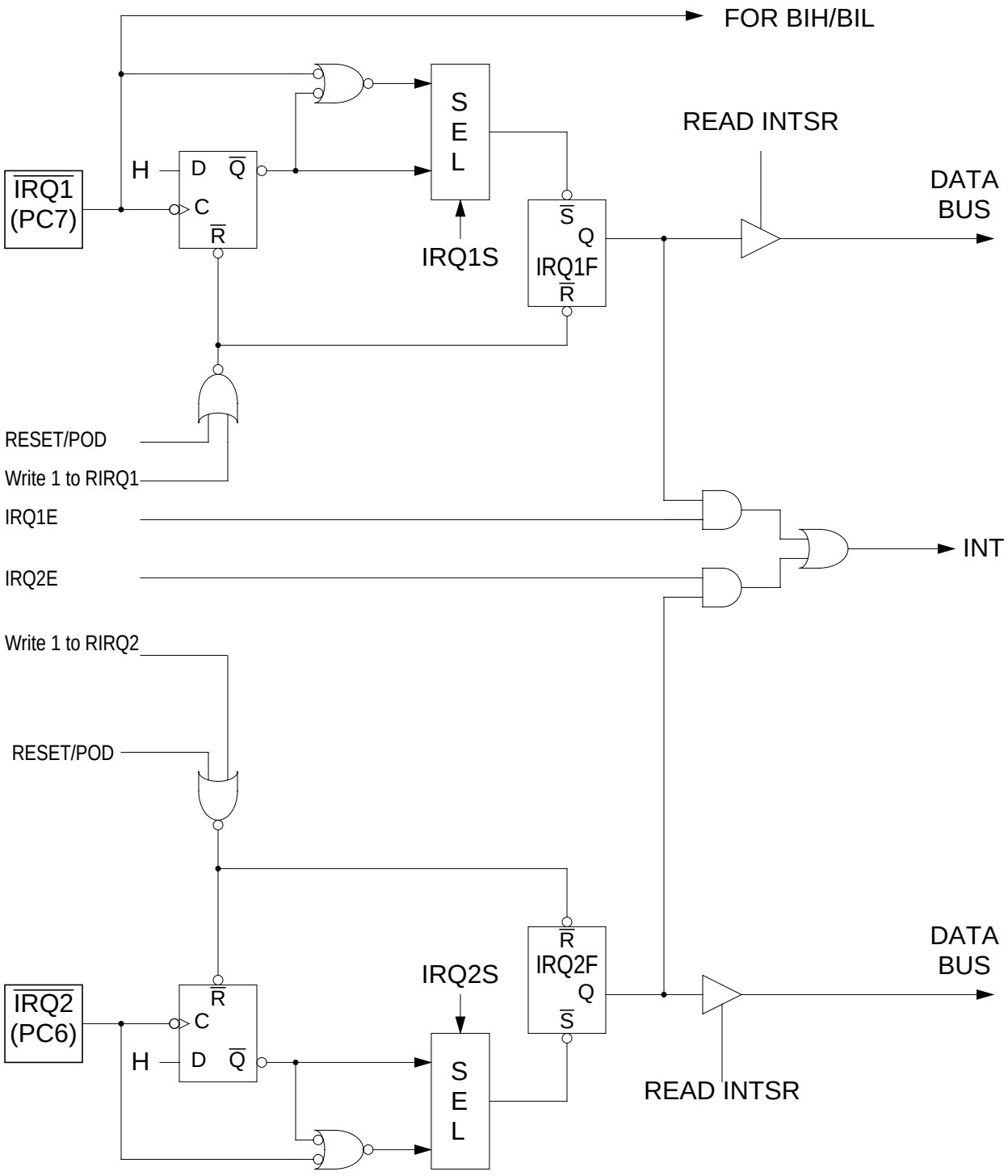


Figure 5-2: IRQ1 and IRQ2 Block Diagram

5.2.2 KEY WAKEUP INTERRUPT (KWI)

There are eight key wakeup inputs (KWI0-7) which share pins with port B. Each key wakeup input is enabled by a corresponding bit in the KWIEN register which resides in the options map, and key wakeup interrupt (KWI) is enabled by the KWIE bit in the INTCR.

When a falling edge is detected at one of the enabled key wakeup inputs, the KWIF bit in the INTSR is set and KWI is generated if KWIE = 1. Each input has a latch which responds only to the falling edge at the pin, and all input latches are cleared at the same time by clearing KWIF bit. Refer to **Figure 5-3: Key Wakeup Interrupt (KWI)**.

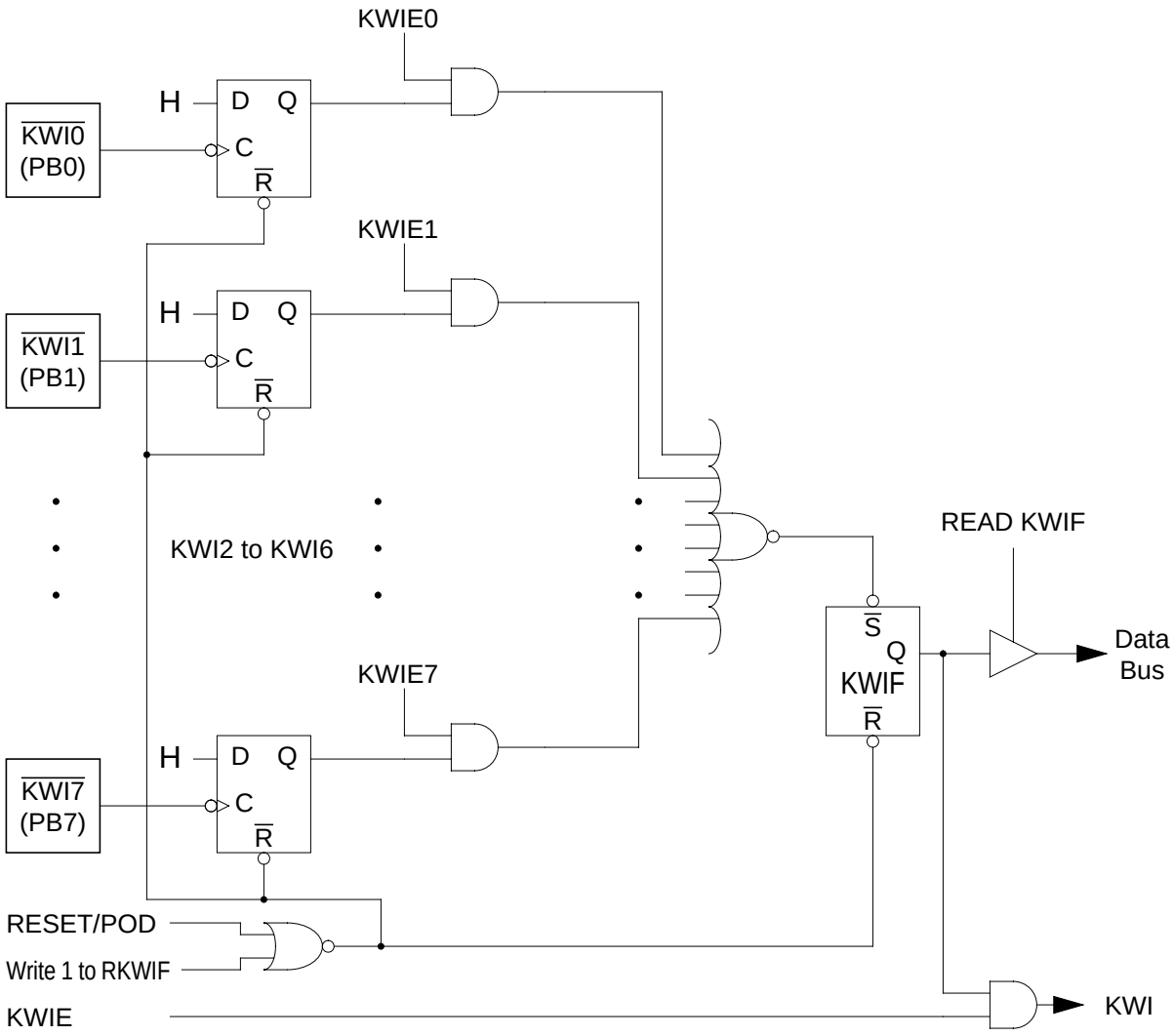


Figure 5-3: Key Wakeup Interrupt (KWI)

5.2.3 KEY WAKE-UP INTERRUPT TIMING

A KWI interrupt request is internally latched and synchronized into the KWI circuit immediately following the falling edge of the $\overline{\text{KWI}}$ source. If KWIE is set, following a delay of one CPU cycle, it is latched into the CPU. If KWIE is not set, the KWI interrupt will be pending until KWIE is set and then latched into the CPU one cycle later. If the interrupt mask bit (I bit) is cleared, the KWI interrupt service routine, specified by the contents of $3FF8:9, will be executed immediately after being latched by the CPU.

> NOTE: If the KWIE is set while a KWI is pending, this interrupt is serviced one instruction cycle following the register update. It is thus advised to code as follows:
>
> ```
> 1) BSET KWIE,INTCR  turn on KWI interrupt
>
> 2) NOP              dummy instruction cycle
>
> 3) ( next instruction intended )
> ```

If a KWI interrupt is pending when the above code sequence is executed, instruction 1) will enable the KWI interrupt, the KWI interrupt will be latched into the CPU during instruction 2) and the KWI interrupt service routine will be executed immediately before instruction 3).

5.2.4 TIMER 1 INTERRUPT

Three timer 1 interrupts (TOI, ICI, and OC1I) share the same interrupt vector at $FFF6 and $FFF7. For more information, refer to **8.1 TIMER 1**.

5.2.5 TIMER 2 INTERRUPT

Two timer 2 interrupts (TI2I and OC2I) share the same interrupt vector at $FFF4 and $FFF5. For more information, see **8.2 TIMER 2**.

5.2.6 SPI1 AND SPI2 INTERRUPTS

Two SPI (SPI1 and SPI2) transfer complete interrupts share the same interrupt vector at $FFF2 and $FFF3. For more information, see **SECTION 7 SERIAL PERIPHERAL INTERFACE (SPI)**.

5.2.7 TB INTERRUPT

The time base interrupt uses the vector at $FFF0 and $FFF1. For more information, see **1.4.1.6 TIME BASE**.

5.2.8 INTERRUPT CONTROL REGISTER (INTCR)

	B7	B6	B5	B4	B3	B2	B1	B0	
$0008	IRQ1E	IRQ2E	0	KWIE	IRQ1S	IRQ2S	0	0	INTCR
RESET:	0	0	0	0	0	0	0	0	

READ: **anytime**

WRITE: **anytime**

IRQ1E **IRQ1 Interrupt Enable**

IRQ1E bit enables IRQ1 interrupt when IRQ1F is set. This bit is cleared at reset.

0 - IRQ1 interrupt is disabled.
1 - IRQ1 interrupt is enabled.

IRQ2E **IRQ2 Interrupt Enable**

IRQ2E bit enables IRQ2 interrupt when IRQ2F is set.
This bit is cleared at reset.

0 - IRQ2 Interrupt is disabled.
1 - IRQ2 Interrupt is enabled.

BIT 5 **Reserved**

This bit is not used and always read as zero.

KWIE **Key Wakeup Interrupt (KWI) Enable**

KWIE bit enables key wakeup interrupt when KWIF is set.
This bit is cleared at reset.

0 - KWI is disabled
1 - KWI is enabled

IRQ1S **IRQ1 Select Edge-Sensitive Only**

0 - IRQ1 is configured for low-level and negative-edge sensitive.
1 - IRQ1 is configured to respond only to negative edges.

IRQ2S **IRQ2 Select Edge Sensitive Only**

0 - IRQ1 is configured for low-level and negative-edge sensitive
1 - IRQ1 is configured to respond only to negative edges.

BITS 1-0 **Reserved**

These bits are not used and always read as zero.

5.2.9 INTERRUPT STATUS REGISTER (INTSR)

	B7	B6	B5	B4	B3	B2	B1	B0	
$0009	IRQ1F	IRQ2F	0	KWIF	RIRQ1	RIRQ2	0	RKWIF	INTSR
RESET:	0	0	0	0	0	0	0	0	

READ: anytime
(Bits 3-0 are write-only bits and always read as zero.)

WRITE: anytime
(Bits 7-4 are read-only bits and write has no effect.)

IRQ1F IRQ1 Interrupt Flag
When IRQ1S = 0, the falling edge or low level at IRQ1 pin sets IRQ1F. When IRQ1S = 1, only the falling edge at the pin sets IRQ1F bit. If IRQ1E bit and this bit are set, interrupt is generated. This bit is a read-only bit and cleared by writing a one to the RIRQ1 bit. Reset clears this bit.

IRQ2F IRQ2 Interrupt Flag
When IRQ2S = 0, the falling edge or low level at IRQ2 pin sets IRQ1F. When IRQ2S = 1, only the falling edge at the pin sets IRQ2F bit. If IRQ1E bit and this bit are set, interrupt is generated. This bit is a read-only bit and is cleared by writing a one to the RIRQ2 bit. Reset clears this bit.

BIT 5 Reserved
This bit is not used and always read as zero.

KWIF Key Wakeup Interrupt Flag
When KWIE*x* bit in the KWIEN register is set, the falling edge at KWI*x* pin sets KWIF bit. If KWIE bit and this bit are set, interrupt is generated. This bit is a read-only bit and clearing KWIF is accomplished by writing a one to the RKWIF bit. Reset clears this bit.

RIRQ1 Reset IRQ1 Flag
The RIRQ1 bit is a write-only bit and always read as zero. Writing a one to this bit clears IRQ1F bit and writing zero to this bit has no effect.

RIRQ2 Reset IRQ2 Flag
The RIRQ2 bit is a write-only bit and always read as zero. Writing a one to this bit clears the IRQ2F bit and writing zero to this bit has no effect.

BIT 1 Reserved
This bit is not used and always read as zero.

RKWIF Reset KWI Flag
The RKWIF bit is a write-only bit and always read as zero. Writing a one to this bit clears KWIF bit and writing zero to this bit has no effect.

SECTION 6 INPUT/OUTPUT PORTS

The MC68HC05G3 (705G4) has eight 8-bit ports and one 4-bit port. Most of these 68 input/output (I/O) pins serve multiple purposes depending on the configuration of the MCU system. The configuration in turn is controlled by hardware mode selection as well as several internal control registers.

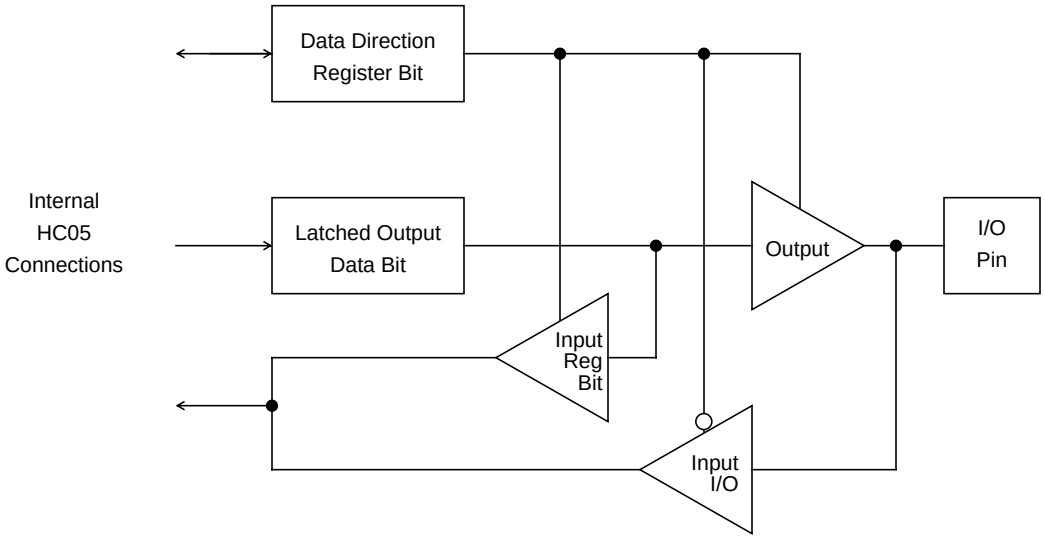


Figure 6-1: Port I/O Circuitry for One Bit

6.1 PORT A

Port A is an 8-bit bidirectional general-purpose port. The data direction of a port A pin is determined by its corresponding DDRA bit.

When a bit is programmed as an output by the corresponding DDRA bit, a data in the PORTA data register becomes an output data to the pin and it is returned for CPU read of PORTA register.

Open drain or CMOS outputs are selected by AWOMH and AWOML bits in the WOM1 register. If the AWOMH bit is set, the P-channel drivers of output buffers of bit 7 through bit 4 are disabled (open drain). If the AWOML bit is set, the P-channel drivers of the output buffers of bit 3 through bit 0 are disabled (open drain).

When a bit is programmed as input by the corresponding DDRA bit, the pin level is read by the CPU.

Port A has pullup resistors as an option. When the RAH or RAL bit in the RCR1 is set, the pullup resistors are attached to the upper four bits or lower four bits of port A pins. (The typical resistor values are to be 50 KΩ @ 3 V.) When a pin outputs a low level, the pullup resistor is disconnected regardless of the state of the RAH or RAL bits.

6.1.1 PORT A DATA REGISTER (PORTA)

	B7	B6	B5	B4	B3	B2	B1	B0	
$0000	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0	PORTA
RESET:	U	U	U	U	U	U	U	U	

READ: anytime
(returns pin level if DDR set to input; returns output data latch if DDR set to output)

WRITE: anytime
(data stored in an internal latch; drives pin only if DDR set for output)

RESET: becomes high impedance inputs

6.1.2 PORT A DATA DIRECTION REGISTER (DDRA)

	B7	B6	B5	B4	B3	B2	B1	B0	
option map $0000	DDRA7	DDRA6	DDRA5	DDRA4	DDRA3	DDRA2	DDRA1	DDRA0	DDRA
RESET:	0	0	0	0	0	0	0	0	

READ: anytime (when OPTM = 1)

WRITE: anytime (when OPTM = 1)

RESET: cleared to $00 (all general-purpose I/O configured for input)

DDRA*x* **Port A Data Direction Register Bit *x***

0 - configure I/O pin PA*x* to input
1 - configure I/O pin PA*x* to output

6.2 PORT B

Port B pins serve two basic functions: key wakeup interrupt (KWI) input pins and general-purpose input pins.

Each KWI input is enabled or disabled by the corresponding KWIE*x* bit in the KWIEN register, and the usage of the KWI input does not affect the general-purpose input function.

Port B pin states may be read any time regardless of the configurations. Since port B has no output drive logic associated with it, there is no DDRB register and the write to the PORTB register has no meaning.

The pullup resistors are provided for both the upper and lower four bits of port B pins which are controlled by the RBH and RBL bits in the RCR1 register. (The typical resistor values are to be 50 KΩ @ 3 V.)

6.2.1 PORT B DATA REGISTER (PORTB)

	B7	B6	B5	B4	B3	B2	B1	B0	
$0001	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0	PORTB
RESET:	U	U	U	U	U	U	U	U	

READ: anytime (returns pin level)

WRITE: has no meaning or effect

RESET: unaffected; always input port

6.3 PORT C

Port C pins share functions with several on-chip peripherals. A pin function is controlled by the enable bit of each associated peripheral.

PC7 and PC6 are general-purpose I/O pins and $\overline{IRQ}$ input pins. The DDRC7/6 bits determine whether the pin states or data latch states should be read by the CPU. When DDRC7/6 =1, the pins become open drain outputs and the $\overline{IRQ}$*x*F can be set by the data latch. Therefore, be sure to clear the flag by software before the $\overline{IRQ}$*x*E bit is enabled.

The PC5 pin is a general-purpose I/O pin and the direction of the pin is determined by the DDRC5 bit in data direction register C (DDRC). When the event output (EVO) is enabled, PC5 is configured as an event output pin and the DDRC5 bit has meaning only for the read of PC5 bit in the PORTC register. If the DDRC5 is set, the PC5 data latch is read by the CPU; otherwise, the PC5 pin level (EVO state) is read. When EVO is disabled, the DDRC5 bit decides the idling state of EVO (if DDRC5 = 1). This PC5/EVO output has the capability to drive a 10 mA source current when (Voh $\geq V_{DD}$ - 0.8 V).

The PC4 and PC3 pins share functions with the timer input pins (EVI and TCAP). These bits are not affected by the usage of timer input functions, and the directions of pins are always controlled by the DDRC4 and DDRC3 bits. Also, the DDRC4 and DDRC3 bits determine whether the pin states or data latch states should be read by the CPU.

The PC2 through PC0 pins are shared with the serial peripheral interface (SPI1). When the SPI1 is not used (SPE1 = 0), DDRC2 through DDRC0 bits control the directions of the pins, and when the SPI1 is enabled, the pins are configured as serial clock output or input (SCK1), serial data output (SDO1), and serial data input (SDI1). The direction of the SCK1 depends on the MSTR1 bit in the SPCR1. The DDRC2 through DDRC0 bits always affect the CPU read of PORTC register (pin states for the input configuration or data latch for the output configuration).

Each port C pin has a pullup resistor option controlled by the corresponding RCR2 register bit. (The typical resistor values are to be 10 KΩ @ 3 V.) When a pin outputs low, the resistor is disconnected regardless of an RCR2 register bit being set.

Bit 5 through bit 0 have open drain or CMOS output options, which are controlled by the corresponding WOM2 register bits. Bits 7 and 6 have fixed open drain outputs. These open drain or CMOS output options are effective to either the general-purpose outputs or the peripheral outputs (EVO, SCK1, and SDO1).

6.3.1 PORT C DATA REGISTER (PORTC)

	B7	B6	B5	B4	B3	B2	B1	B0	
$0002	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0	PORTC
RESET:	U	U	U	U	U	U	U	U	

READ: anytime
(returns pin level if DDR set to input; returns output data latch if DDR set to output)

WRITE: anytime
(data stored in an internal latch; drives pin only if DDR set for output writes; do not change pin state when pin configured for peripheral output for SDO1, SCK1, and EVO.)

RESET: becomes high impedance inputs

6.3.2 PORT C DATA DIRECTION REGISTER (DDRC)

option map $0002	B7	B6	B5	B4	B3	B2	B1	B0	
	DDRC7	DDRC6	DDRC5	DDRC4	DDRC3	DDRC2	DDRC1	DDRC0	DDRC
RESET:	0	0	0	0	0	0	0	0	

READ: anytime (when OPTM = 1)

WRITE: anytime (when OPTM = 1)

RESET: cleared to $00 (all general-purpose I/O configured for input)

DDRC*x* **Port C Data Direction Register Bit *x***

0 - configure I/O pin PC*x* to input
1 - configure I/O pin PC*x* to output

The timer and SPI1 force the I/O state to be an output for each port C line associated with an enabled output function such as SDO1 and EVO. In this case, the data direction bits will not change. If bit 7 or bit 6 is enabled, the corresponding port bit always becomes an open drain output.

6.4 PORT D

Port D is an 8-bit bidirectional general-purpose port. The data direction of a port D pin is determined by its corresponding DDRD bit.

When a bit is programmed as an output by the corresponding DDRD bit, data in the PORTD data register becomes output data to the pin and it is returned for CPU read of PORTD register.

When a bit is programmed as input by the corresponding DDRD bit, the pin level is read by the CPU.

6.4.1 PORT D DATA REGISTER (PORTD)

	B7	B6	B5	B4	B3	B2	B1	B0	
$0003	PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0	PORTD
RESET:	U	U	U	U	U	U	U	U	

READ: anytime
(returns pin level if DDR set to input; returns output data latch if DDR set to output)

WRITE: anytime
(data stored in an internal latch; drives pin only if DDR set for output)

RESET: becomes high impedance inputs

6.4.2 PORT D DATA DIRECTION REGISTER (DDRD)

option map $0003	B7	B6	B5	B4	B3	B2	B1	B0	
	DDRD7	DDRD6	DDRD5	DDRD4	DDRD3	DDRD2	DDRD1	DDRD0	DDRD
RESET:	0	0	0	0	0	0	0	0	

READ: anytime (when OPTM = 1)

WRITE: anytime (when OPTM = 1)

RESET: cleared to $00 (all general-purpose I/O configured for input)

DDRD*x* **Port D Data Direction Register Bit *x***

0 - configure I/O pin PD*x* to input
1 - configure I/O pin PD*x* to output

6.5 PORT E

Port E is an 8-bit bidirectional general-purpose port. The data direction of a port E pin is determined by its corresponding DDRE bit.

When a bit is programmed as an output by the corresponding DDRE bit, data in the PORTE data register becomes output data to the pin and it is returned for CPU read of PORTE register.

When a bit is programmed as input by the corresponding DDRE bit, the pin level is read by the CPU.

6.5.1 PORT E DATA REGISTER (PORTE)

	B7	B6	B5	B4	B3	B2	B1	B0	
$0004	PE7	PE6	PE5	PE4	PE3	PE2	PE1	PE0	PORTE
RESET:	U	U	U	U	U	U	U	U	

READ: anytime
(returns pin level if DDR set to input; returns output data latch if DDR set to output)

WRITE: anytime
(data stored in an internal latch; drives pin only if DDR set for output)

RESET: becomes high impedance inputs

6.5.2 PORT E DATA DIRECTION REGISTER (DDRE)

option map $0004	B7	B6	B5	B4	B3	B2	B1	B0	
	DDRE7	DDRE6	DDRE5	DDRE4	DDRE3	DDRE2	DDRE1	DDRE0	DDRE
RESET:	0	0	0	0	0	0	0	0	

READ: anytime (when OPTM = 1)

WRITE: anytime (when OPTM = 1)

RESET: cleared to $00 (all general-purpose I/O configured for input)

DDRE*x* **Port E Data Direction Register Bit *x***

0 - configure I/O pin PE*x* to input
1 - configure I/O pin PE*x* to output

6.6 PORT F

Port F pins serve two basic functions: A/D converter input pins and general-purpose input pins. See **10.4.4 CH3:CH0 - CHANNEL SELECT BITS**.

Since no output drive logic is associated with port F, there is no DDRF register and the write to the PORTF register has no meaning.

6.6.1 PORT F DATA REGISTER (PORTF)

	B7	B6	B5	B4	B3	B2	B1	B0	
$0005	PF7	PF6	PF5	PF4	PF3	PF2	PF1	PF0	PORTF
RESET:	U	U	U	U	U	U	U	U	

READ: anytime (returns pin level)

WRITE: has no meaning or effect

RESET: unaffected; always input port

6.7 PORT G

Port G pins share the functions with several on-chip peripherals. A pin function is controlled by the enable bit of each associated peripheral.

PG7 through PG4 are general-purpose I/O pins and PWM output pins. When the PWM is enabled, one or more of the channels, PG7 through PG4, will be configured as a PWM output pin regardless of the state of DDRG7 through DDRG4. The DDRG7 through DDRG4 bits determine the CPU read of the PORTG register (pin states for the input configuration or data latch for the output configuration).

The PG3 pin shares function with the timer output pin (TCMP). When PG3 is configured as an output, it will be tied to the TCMP and cannot be used to provide output from the data register. The PG3 pin state always will be read by the CPU, regardless of the state of DDRG3.

The PG2 through PG0 pins are shared with the serial peripheral interface (SPI2). When the SPI2 is not used (SPE2 = 0), DDRG2 through DDRG0 bits control the directions of the pins, and when the SPI2 is enabled, the pins are configured as serial clock output or input (SCK2), serial data output (SDO2), and serial data input (SDI2). The direction of the SCK2 depends on the MSTR2 bit in the SPCR2. The DDRG2 through DDRG0 bits always affect the CPU read of PORTG register (pin states for the input configuration or data latch for the output configuration.)

Open drain or CMOS outputs are selected by GWOMH and GWOML bits in the WOM1 register. If the GWOMH bit is set, the P-channel drivers of output buffers of bit 7 through bit 4 are disabled (open drain). If the GWOML bit is set, the P-channel drivers of output buffers of bit 3 through bit 0 are disabled (open drain). These open drain or CMOS output options are effective to either the general-purpose outputs or the peripheral outputs (PWM, TCMP, SCK2, and SDO2).

Port G has pullup resistors as an option. When the RGH or RGL bit in the RCR1 is set, the pullup resistors are attached to the upper four bits or lower four bits of port G pins. (The typical resistor values are to be 10 KΩ @ 3 V.) When a pin outputs a low level, the pullup resistor is disconnected regardless of the states of the RGH or RGL bits.

6.7.1 PORT G DATA REGISTER (PORTG)

	B7	B6	B5	B4	B3	B2	B1	B0	
$0006	PG7	PG6	PG5	PG4	PG3	PG2	PG1	PG0	PORTG
RESET:	U	U	U	U	U	U	U	U	

READ: anytime
(returns pin level if DDR set to input; returns output data latch if DDR set to output; except for PG3, always return pin level regardless of the state of DDRG3)

WRITE: anytime
(data stored in an internal latch; drives pin only if DDR set for output
writes do not change pin state when pin configured for TCMP, SDO2, SCK2, and PWMs peripheral output for TCMP,SDO2, SCK2, and PWMs)

RESET: becomes high impedance inputs

6.7.2 PORT G DATA DIRECTION REGISTER (DDRG)

option map $0006	B7	B6	B5	B4	B3	B2	B1	B0	
	DDRG7	DDRG6	DDRG5	DDRG4	DDRG3	DDRG2	DDRG1	DDRG0	DDRG
RESET:	0	0	0	0	0	0	0	0	

READ: anytime (when OPTM = 1)

WRITE: anytime (when OPTM = 1)

RESET: cleared to $00 (all general-purpose I/O configured for input) DDRG*x*

Port G Data Direction Register Bit *x*

0 - configure I/O pin PG*x* to input
1 - configure I/O pin PG*x* to output

The PWM and SPI2 force the I/O state to be an output for each port G line associated with an enabled output function such as SDO2 and PWMs. In this case, the data direction bits will not change. When DDRG3 configures PG3 as an output, it will be tied to the TCMP and cannot be used to provide output from the data register.

6.8 PORT H

Port H is an 8-bit bidirectional general-purpose port. The data direction of a port H pin is determined by its corresponding DDRH bit.

When a bit is programmed as an output by the corresponding DDRH bit, data in the PORTH data register becomes output data to the pin and it is returned for CPU read of PORTH register. These outputs have the capability to drive10 mA sink current when (Vol ≤ V_{SS} + 0.8 V).

Open drain or CMOS outputs are selected by HWOMH and HWOML bits in the WOM1 register. If the HWOMH bit is set, the P-channel drivers of output buffers of bit 7 through bit 4 are disabled (open drain). If the HWOML bit is set, the P-channel drivers of output buffers of bit 3 through bit 0 are disabled (open drain).

When a bit is programmed as input by the corresponding DDRH bit, the pin level is read by the CPU.

Port H has pullup resistors as an option. When the RHH or RHL bit in the RCR1 is set, the pullup resistors are attached to the upper four bits or lower four bits of port H pins. (The typical resistor values are to be 50 KΩ @ 3 V.) When a pin outputs a low level, the pullup resistor is disconnected regardless of the states of RHH or RHL bits.

6.8.1 PORT H DATA REGISTER (PORTH)

	B7	B6	B5	B4	B3	B2	B1	B0	
$0007	PH7	PH6	PH5	PH4	PH3	PH2	PH1	PH0	PORTH
RESET:	U	U	U	U	U	U	U	U	

READ: anytime
(returns pin level if DDR set to input; returns output data latch if DDR set to output)

WRITE: anytime (data stored in an internal latch; drives pin only if DDR set for output)

RESET: becomes high impedance inputs

6.8.2 PORT H DATA DIRECTION REGISTER (DDRH)

option map $0007	B7	B6	B5	B4	B3	B2	B1	B0	
	DDRH7	DDRH6	DDRH5	DDRH4	DDRH3	DDRH2	DDRH1	DDRH0	DDRH
RESET:	0	0	0	0	0	0	0	0	

READ: anytime (when OPTM = 1)

WRITE: anytime (when OPTM = 1)

RESET: cleared to $00 (all general-purpose I/O configured for input)

DDRH*x* **Port H Data Direction Register Bit *x***

0 - configure I/O pin PH*x* to input
1 - configure I/O pin PH*x* to output

6.9 PORT J

Port J is a 4-bit general-purpose output port. Any read from port J will return the output data latch. Since no input drive logic is associated with port J, there is no DDRJ register.

6.9.1 PORT J DATA REGISTER (PORTJ)

	B7	B6	B5	B4	B3	B2	B1	B0	
$003C	0	0	0	0	PJ3	PJ2	PJ1	PJ0	PORTJ
RESET:	0	0	0	0	0	0	0	0	

READ: anytime (returns output data latch; bit 4 through 7 are not used)

WRITE: data stored in an internal latch

RESET: cleared to $00; always output port

SECTION 7 SERIAL PERIPHERAL INTERFACE (SPI)

Two serial peripheral interfaces (SPI) are built into the MC68HC05G3 (705G4).

In the SPI format, three separate wires are required for data input, output, and clock. In this format, the clock is not being included in the data stream and must be provided as a separate signal. The three pins are occupied for serial clock, input data, and output data.

When one of the SPIs is enabled (SPE = 1), bit 0 through bit 2 of port C/port G become SDI, SDO, and SCK pins, and the corresponding DDRC/DDRG bit has no affect on the direction of the pin.

The MSTR bit decides the SPI operation mode; SCK pin is configured as output in the master mode and configured as input in the slave mode.

The DORD bit in the serial peripheral control register (SPCR) selects the data transmission order. When DORD bit is set, the LSB of serial data is shifted out/in first. When the DORD bit is clear, serial data is shifted from MSB.

Serial clock speed is selectable by the SPR bit in the SPCR. Interrupt may be generated by the completion of transfer.

7.1 FEATURES

- Full Duplex, Three-Wire Synchronous Transfers
- Master and Slave Operation
- Programmable Data Transmission Order
- E/2 (Maximum) Master Bit Frequency
- E (Maximum) Slave Bit Frequency
- Two Programmable Master Bit Rates
- End of Transmission Interrupt Flag
- Wakeup from Stop Mode (Slave Mode Only)

7.2 FUNCTIONAL DESCRIPTIONS

A block diagram of the serial peripheral is shown in **Figure 7-2**.

The clock start logic is triggered by CPU (detection of CPU write to the 8-bit shift register (SPDR)) and originates the system clock (SCK) based on the internal processor clock. This clock also is used in the 3-bit counter and 8-bit shift register.

After data is written to the 8-bit shift register of the master device, it is then shifted out to the SDO pin for application to the slave device. At the same time, data applied from a slave device via the SDI pin is shifted into the 8-bit shift register.

After 8-bit data is shifted in/out, SCK stops and SPIF is set, and if SPIE is enabled, an interrupt request is generated. The slave device in the stop mode wakes up by this interrupt. Further transfer (write to SPDR) is inhibited while SPIF is 1.

Figure 7-1: SPI Master-Slave Interconnection illustrates the master-slave basic interconnection.

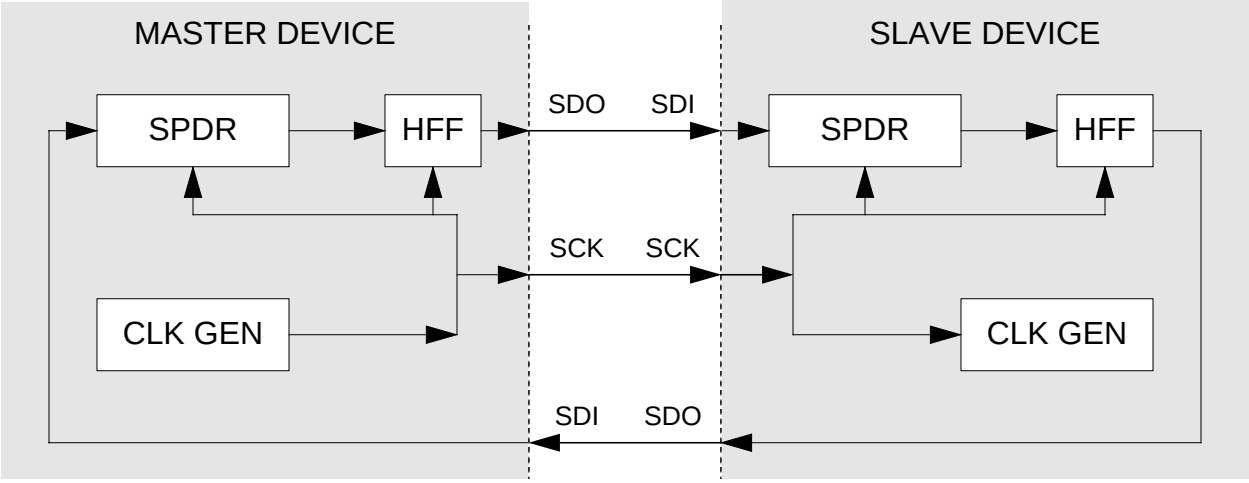


Figure 7-1: SPI Master-Slave Interconnection

7.2.1 INTERNAL BLOCK DESCRIPTIONS

This section describes the main blocks in the SPI module.

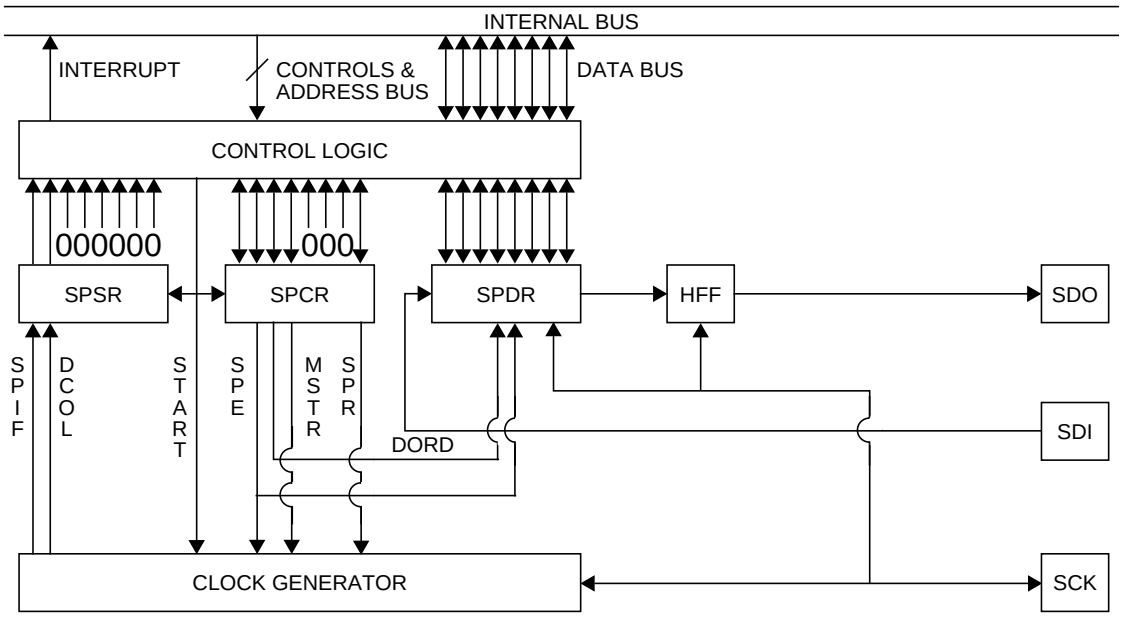


Figure 7-2: SPI Block Diagram

CONTROL	This block is an interface to the HC05 internal bus and generates a START signal when writing to SPDR is detected in the master mode. It also generates an interrupt request to the CPU.
SPDR	This register is an 8-bit shift register called serial peripheral data register (SPDR). The DORD bit in the SPCR determines the bus connection between internal data bus and SPDR. This register can be read and written by the CPU.
SPCR	Serial peripheral control register (SPCR). The bits in this register control SPI functions.
SPSR	Serial peripheral status register (SPSR). This register mainly sets flags such as SPIF and DCOL.
CLKGEN	In the master mode, this block generates serial clock (SCK) when CPU writes to data register (SPDR) and the clock rate is selected by SPR bit in the control register. In slave mode, external clock from SCK pin is used instead of master mode clock and SPR has no affect on SCK. This clock generator includes a 3-bit clock counter. Overflow of this counter sets SPIF.

7.3 SIGNAL DESCRIPTIONS

Three basic signals (SDI, SDO, and SCK) are described in the following paragraphs. **Figure 7-3: Clock-Data Timing Diagram** shows the relationship among SCK, SDI, and SDO.

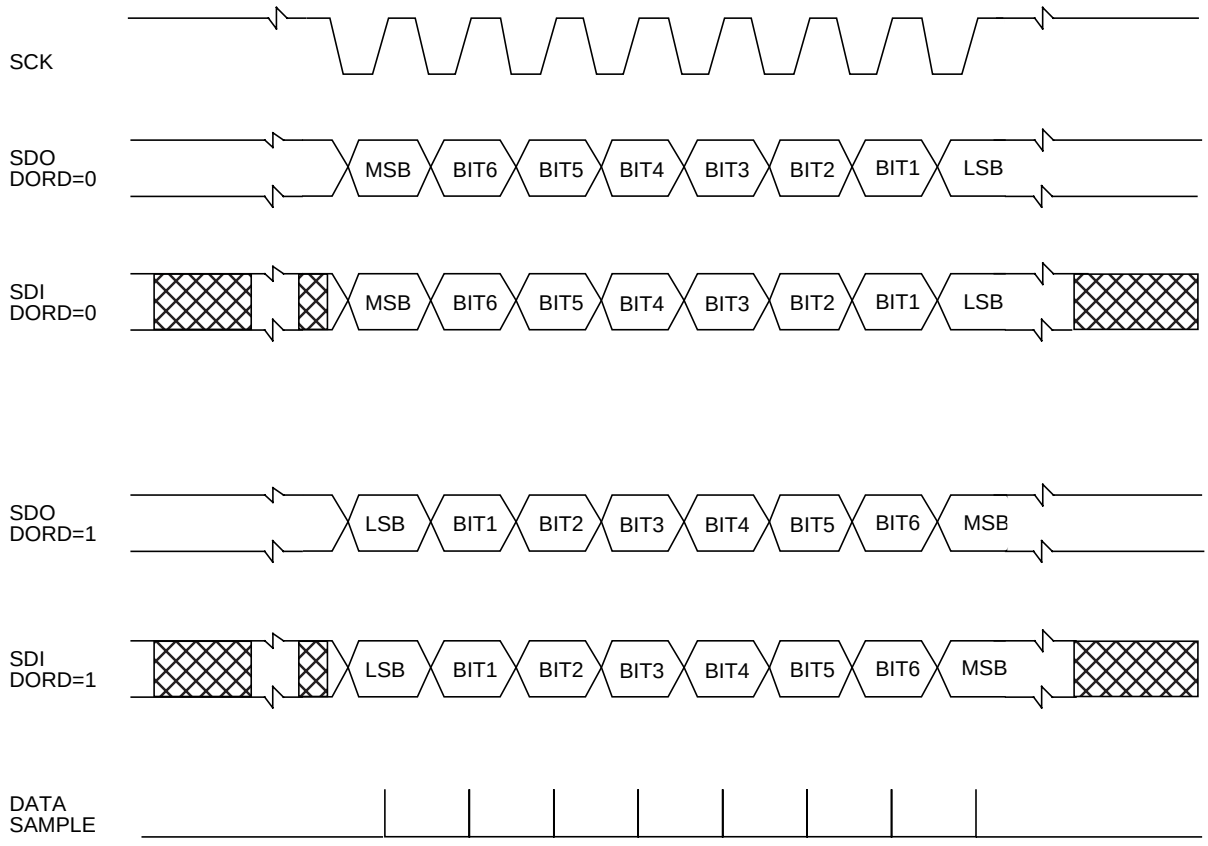


Figure 7-3: Clock-Data Timing Diagram

7.3.1 SPI DATA I/O (SDI and SDO)

Two serial data lines, SDI for input and SDO for output, are connected to I/O port when SPI is enabled (SPE=1).

At the falling edge of SCK, a serial data bit is transmitted out of the SDO pin. At the rising edge of SCK, a serial data bit on the SDI pin is sampled internally.

When data is transmitted to other devices via the SDO line, the receiving data comes into the shift register through the SDI pin. This implies full duplex transmission with both data-out and data-in synchronized with the same clock signal. Thus, the byte transmitted is replaced by the byte received and eliminates the need for separate transmit-empty and receiver-full status bits. A single status bit, SPIF, is used to signify the completion of data transfer.

7.3.2 SERIAL CLOCK (SCK)

The serial clock (SCK) is used for synchronization of both the input and output data streams through its SDI and SDO pins. PC3(SCK) should be high before the SPI is enabled. This can be done with the internal resistor or an external resistor, or by setting DDRC3=1 and PC3=1.

The master and slave devices are capable of exchanging a data byte during a sequence of eight clock pulses. Since the SCK is generated by the master/slave, data transfer is accomplished by synchronization to SCK.

The master generates the SCK through a circuit driven by the internal processor clock, and uses the SCK to latch incoming slave device data on the SDI pin and to shift out data to the slave via the SDO pin. The SPR bit in the SPCR of the master selects the clock rate.

The slave device receives the SCK from the master device and uses the SCK to latch incoming master device data on the SDI pin and to shift out data to the master via the SDO pin. The SPR bit in the SPCR of the slave has no meaning.

7.4 REGISTERS

Three registers in each serial peripheral interface (SPI) provide control, status, and data storage functions. These three registers are the serial peripheral control register (SPCR location $000A/$000D), serial peripheral status register (SPSR location $000B/$000E), and serial peripheral data register (SPDR location $000C/$000F).

7.4.1 SERIAL PERIPHERAL CONTROL REGISTER (SPCR*x*)

	B7	B6	B5	B4	B3	B2	B1	B0	
$000A	SPIE1	SPE1	DORD1	MSTR1	0	0	0	SPR1	SPCR1
RESET:	0	0	0	0	0	0	0	0	

	B7	B6	B5	B4	B3	B2	B1	B0	
$000D	SPIE2	SPE2	DORD2	MSTR2	0	0	0	SPR2	SPCR2
RESET:	0	0	0	0	0	0	0	0	

READ: anytime

WRITE: should not write during transmission

SPIE*x* **SPI Interrupt Enable**

If the serial peripheral interrupt enable (SPIE*x*) bit is set, interrupt is generated when SPIF*x* in the SPSR*x* is set and the I bit (interrupt mask bit) in the condition code register (CCR) is clear.

In stop mode, serial peripheral interrupt request is accepted only in the slave mode. Interrupt in the master mode will be pending until stop mode is exited. STOP instruction does not change SPIF*x* or SPIE*x*.

0 - disable SPI interrupt
1 - enable SPI interrupt

SPE*x* **SPI Enable**

When the serial peripheral interface enable (SPE*x*) bit is set, the SPI system is enabled and connected to the port C/port G pins.

Clearing SPE*x* bit initializes all control logic in the SPI*x* modules and disconnects the SPI*x* from port C/port G pins.

This bit is cleared at reset.

0 - disable SPI
1 - enable SPI

DORD*x* **Data transmission ORDer**

When this bit is set, the data in the 8-bit shift register (SPDR*x*) is shifted in/out from the LSB. When this bit is cleared, the data in the SPDR*x* is shifted in/out from MSB.

This bit is cleared at reset.

0 - MSB first
1 - LSB first

MSTR*x* **MaSTeR mode select**

The MSTR*x* bit determines whether the device is in master mode or slave mode.

In the master mode (MSTR*x* = 1), SCK*x* pin is configured as the output pin and the serial clock is generated by the internal clock generator when the CPU writes to the SPDR.

In the slave mode (MSTR*x* = 0), SCK*x* pin is configured as the input pin and the serial clock is applied externally.

This bit is cleared at reset.

0 - slave mode
1 - master mode

BITS 3-1 **Reserved**

These bits are not used and are fixed to zero.

SPR*x* **SPI*x* clock rate select**

This serial peripheral clock rate bit selects one of two bit rates of SCKx. This bit is cleared at reset.

0 - Internal processor clock divided by 2
1 - Internal processor clock divided by 16

7.4.2 SERIAL PERIPHERAL STATUS REGISTER (SPSR*x*)

	B7	B6	B5	B4	B3	B2	B1	B0	
$000B	SPIF1	DCOL1	0	0	0	0	0	0	SPSR1
RESET:	0	0	0	0	0	0	0	0	

	B7	B6	B5	B4	B3	B2	B1	B0	
$000E	SPIF2	DCOL2	0	0	0	0	0	0	SPSR2
RESET:	0	0	0	0	0	0	0	0	

READ: anytime

WRITE: not applicable

SPIF*x* **Serial transfer complete flag**

The serial peripheral data transfer complete flag bit notifies the user that a data transfer between the MC68HC05G3 (705G4) and external device has been completed. With the completion of the data transfer, the rising edge of the eighth pulse set SPIF*x*, and if SPIE*x* is set, serial peripheral interrupt (SPI*x*) is generated. However, during stop mode, interrupt request is serviced only in slave mode. STOP execution never affects this SPIF*x* flag or SPIE*x*.

When SPIF is set, the ninth clock from the clock generator or from the SCK pin is inhibited.

Clearing the SPIF*x* bit is accomplished by a software sequence of accessing the SPSR*x* while the SPIF*x* bit is set, followed by accessing the SPDR*x* (8-bit shift register). While SPIF*x* is set, all writes to the SPDR*x* are inhibited until SPSR*x* is read by the CPU.

SPIF*x* bit is a read-only bit and cleared by reset.

0 - data transfer not complete
1 - data transfer complete

DCOL*x* **Data COLlision**

The data collision bit notifies the user that an attempt was made to write or read the serial peripheral data register while a data transfer was taking place with an external device. The transfer continues uninterrupted; therefore, a write will be unsuccessful, and a data read may be incorrect.

A “data collision” only sets the DCOL*x* bit and does not generate SPI*x* interrupt. The DCOL*x* bit indicates only the occurrence of data collision.

Clearing the DCOL*x* bit is accomplished by a software sequence of accessing the SPSR*x* while SPIF*x* is set, followed by accessing the SPDR*x*. Both SPIF*x* and DCOL*x* bits will be cleared by this sequence.

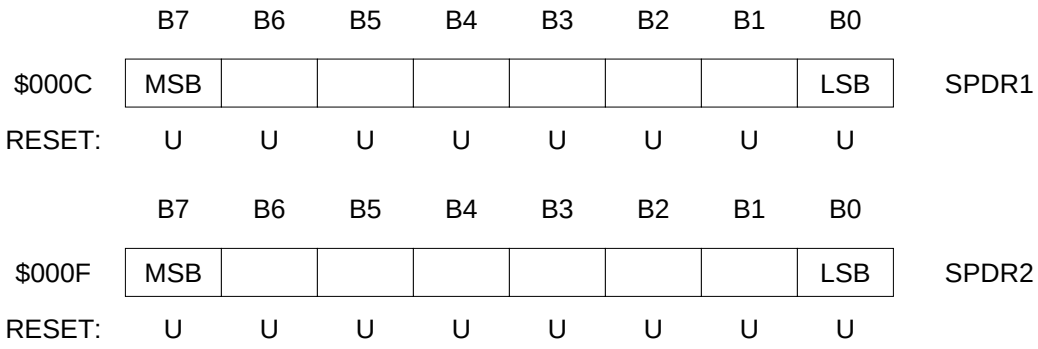
The DCOL*x* bit is cleared by reset.

0 - no data collision
1 - data collision occurred

BITS 5-0 **Reserved**

These bits are not used and are fixed to zero.

7.4.3 SERIAL PERIPHERAL DATA REGISTER (SPDR*x*)



READ: A read during transmission causes DCOL*x* to be set.

WRITE: A write during transmission causes DCOL*x* to be set.

The serial peripheral data register (SPDR*x*) is used to transmit and receive data on the serial bus.

In master mode, a write to this register initiates transmission/reception of data byte.

At the completion of transmitting a byte data, the SPIF*x* status bit is set. A write to the SPDR*x* is inhibited while this register is shifting (this condition causes DCOL*x* to be set) or when the SPIF*x* bit is set without reading SPSR*x*. “Data collision” never affects the receiving and transmitting data in SPDR*x*.

A write or read of the SPDR*x* after accessing the SPSR*x* with SPIF*x* set will clear SPIF*x* and DCOL*x* bits.

The ability to access SPDR*x* is inhibited when a transmission is taking place. It is important to read the discussion defining DCOL*x* and SPIF*x* bits to understand the limits on using the SPDR*x*.

When serial peripheral interface (SPI*x*) is not used (SPE*x* = 0), this SPDR*x* can be used as a general-purpose data storage register.

7.5 PORT FUNCTION

SPI1 module shares the port with PC0 through PC2 and SPI2 module shares the port with PG0 through PG2.

The SPI1 shares I/O pins with PC0 through PC2. When SPE1 is set, PC0 becomes SDI1 input, PC1 becomes SDO1 output and PC2 becomes SCK1. The direction of SCK1 depends on MSTR1 bit. Setting DDRC bits 0-2 does not change the data direction of the pin to output, but instead changes the source of data when PC0-2 is read. If DDRC*x* = 1, port C bit-*x* data latch is read and if DDRC*x* = 0, PORTC*x* pin level is read by the CPU.

The SPI2 shares I/O pins with PG0 through PG2. When SPE2 is set, PG0 becomes SDI2 input, PG1 becomes SDO2 output and PG2 becomes SCK2. The direction of SCK2 depends on MSTR2 bit. Setting DDRG bits 0-2 does not change the data direction of the pin to output, but instead changes the source of data when PG0-2 is read. If DDRG*x* = 1, port G bit-*x* data latch is read and if DDRG*x* = 0, PORTG*x* pin level is read by the CPU.

When SPE*x* is cleared, SPI*x* is disconnected and PC0 through PC2 (SPI1) or PG0 through PG2 (SPI2) are used as general-purpose I/O pins. For more information on the ports, see **6.3 PORT C,** and **6.7 PORT G**.

THIS PAGE INTENTIONALLY LEFT BLANK

SECTION 8 TIMER SYSTEM

The MC68HC05G3 (705G4) has two timer modules, two timer input pins (TCAP and EVI), and two timer output pins (TCMP and EVO). The following block diagram shows the timer system of the MC68HC05G3 (705G4).

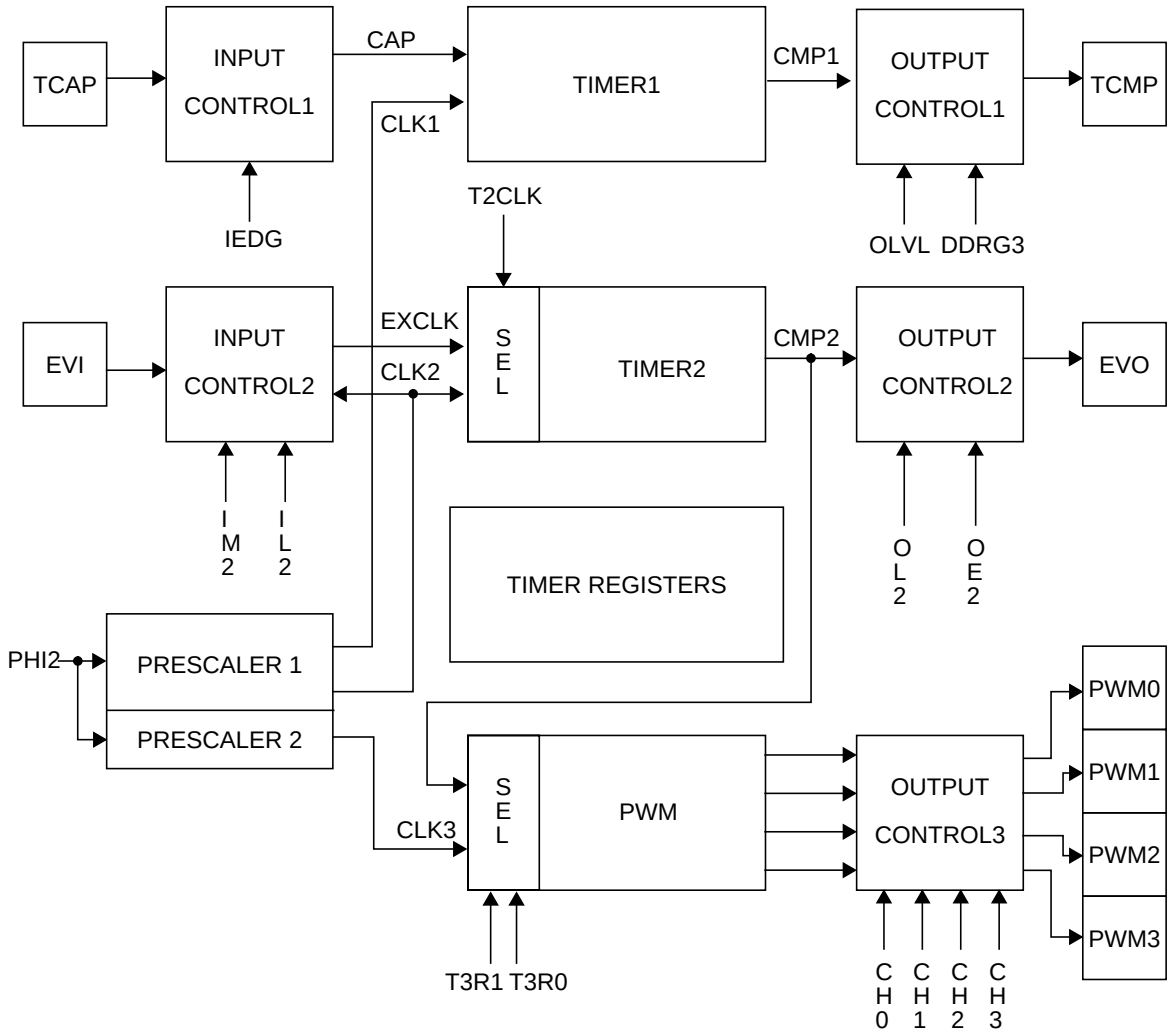


Figure 8-1: Timer Block Diagram

8.1 TIMER 1

Timer 1 is a 16-bit, free-running up counter which has one 16-bit input capture and one 16-bit output compare. The timer is driven by a fixed system clock divided by four.

This timer can be used for many purposes, including input waveform measurements while simultaneously generating an output compare interrupt. Pulse widths can vary from several microseconds to many seconds. Refer to **Figure 8-2: Timer 1 Block Diagram**. Because the timer has a 16-bit architecture, each specific functional segment (capability) is

represented by two registers. These registers contain the high and low bytes of that functional segment. Generally, accessing the low byte of a specific timer function allows full control of that function; however, an access of the high byte inhibits that specific timer function until the low byte also is accessed.

NOTE: The I bit in the CCR should be set while manipulating both the high and low byte registers of a specific timer function to ensure that an interrupt does not occur.

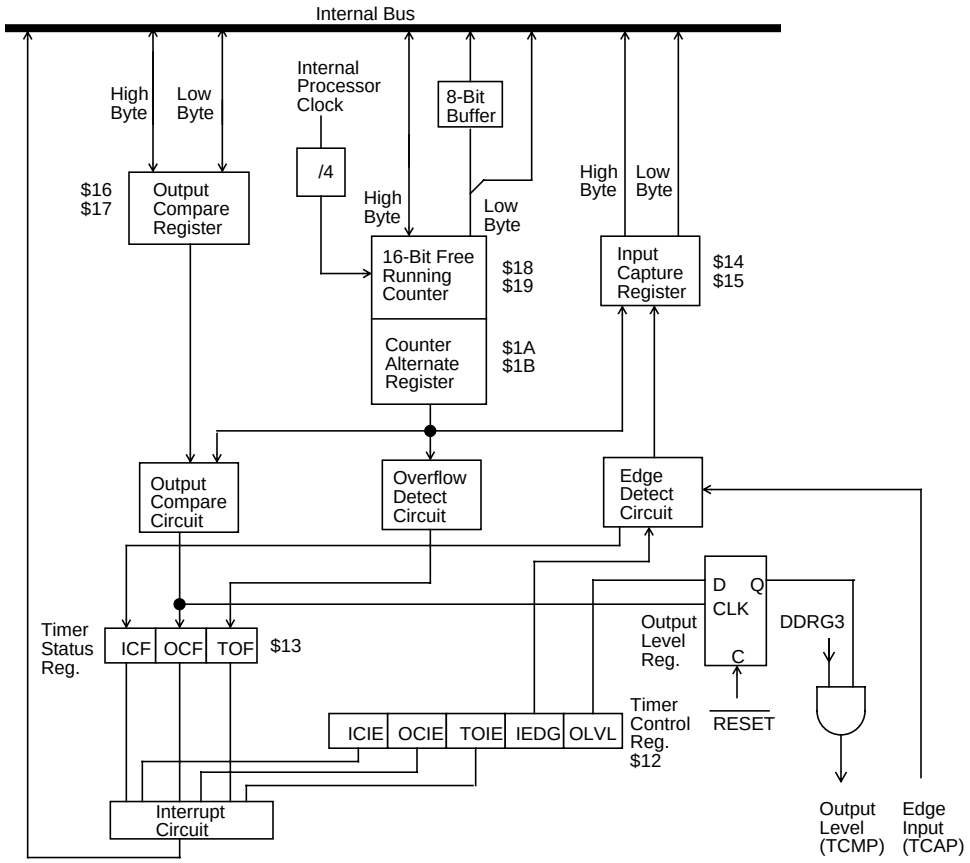


Figure 8-2: Timer 1 Block Diagram

8.1.1 COUNTER

The key element in the programmable timer is a 16-bit, free-running counter or counter register, preceded by a prescaler that divides the internal processor clock by four. The prescaler gives the timer a resolution of 2.0 microseconds if the internal bus clock is 2.0 MHz. The counter is incremented during the low portion of the internal bus clock. Software can read the counter at any time without affecting its value.

The double-byte, free-running counter can be read from either of two locations, $18-$19 (counter register) or $1A-$1B (counter alternate register). A read from only the least significant byte (LSB) of the free-running counter ($19, $1B) receives the count value at the

time of the read. If a read of the free-running counter or counter alternate register first addresses the most significant byte (MSB) ($18, $1A), the LSB ($19, $1B) is transferred to a buffer. This buffer value remains fixed after the first MSB read, even if the user reads the MSB several times. This buffer is accessed when reading the free-running counter or counter alternate register LSB ($19 or $1B) and, thus, completes a read sequence of the total counter value. In reading either the free-running counter or counter alternate register, if the MSB is read, the LSB also must be read to complete the sequence.

The counter alternate register differs from the counter register in one respect: A read of the counter register MSB can clear the timer overflow flag (TOF). Therefore, the counter alternate register can be read at any time without the possibility of missing timer overflow interrupts due to clearing of the TOF.

The free-running counter is configured to $FFFC during reset and always is a read-only register. During a power-on reset, the counter is also preset to $FFFC and begins running after the oscillator start-up delay. Because the free-running counter is 16 bits preceded by a fixed divided-by-four prescaler, the value in the free-running counter repeats every 262,144 internal bus clock cycles. When the counter rolls over from $FFFF to $0000, the TOF bit is set. An interrupt also can be enabled when counter rollover occurs by setting its interrupt enable bit (TOIE).

8.1.2 OUTPUT COMPARE REGISTER

The 16-bit output compare register is made up of two 8-bit registers at locations $16 (MSB) and $17 (LSB). The output compare register is used for several purposes, such as indicating when a period of time has elapsed. All bits are readable and writable and are not altered by the timer hardware or reset. If the compare function is not needed, the two bytes of the output compare register can be used as storage locations.

The output compare register contents are compared with the contents of the free-running counter continually, and, if a match is found, the corresponding output compare flag (OCF) bit is set and the corresponding output level (OLVL) bit is clocked to an output level register. The output compare register values and the output level bit should be changed after each successful comparison to establish a new elapsed timeout. An interrupt also can accompany a successful output compare provided the corresponding interrupt enable bit (OCIE) is set.

After a processor write cycle to the output compare register containing the MSB ($16), the output compare function is inhibited until the LSB ($17) also is written. The user must write both bytes (locations) if the MSB is written first. A write made only to the LSB ($17) will not inhibit the compare function. The free-running counter is updated every four internal bus clock cycles. The minimum time required to update the output compare register is a function of the program rather than the internal hardware.

The processor can write to either byte of the output compare register without affecting the other byte. The output level (OLVL) bit is clocked to the output level register regardless of whether the output compare flag (OCF) is set or clear. A valid output compare must occur before the OLVL bit becomes available at the output compare pin (TCMP) with DDRG3 set.

Because neither the output compare flag (OCF bit) nor output compare register is affected by reset, care must be exercised when initializing the output compare function with software. The following procedure is recommended:

1. Set DDRG3 bit to configure PG3 as an output tied to TCMP.
2. Write to the high byte of the output compare register to inhibit further compares until the low byte is written.
3. Read the timer status register to arm the OCF if it is already set.
4. Write to the low byte of the output compare register to enable the output compare function with the flag clear.

The advantage of this procedure is to prevent the OCF bit from being set between the time it is read and the write to the output compare register. A software example is shown below.

```
10   3E    BSET OPTM,MISC      SWITCH TO OPTION MAP
16   06    BSET DDRG3,DDRG     CONFIGURE PG3 AN OUTPUT
11   3E    BCLR OPTM,MISC      RETURN TO MAIN MAP
B7   16    STA OCMPHI          INHIBIT OUTPUT COMPARE
B6   13    LDA TSTAT           ARM OCF BIT IF SET
BF   17    STX OCMPLD          READY FOR NEXT COMPARE
```

8.1.3 INPUT CAPTURE REGISTER

Two 8-bit registers, which make up the 16-bit input capture register, are read-only and are used to latch the value of the free-running counter after the corresponding input capture edge detector senses a defined transition. The level transition which triggers the counter transfer is defined by the corresponding input edge bit (IEDG). Reset does not affect the contents of the input capture register.

The result obtained by an input capture will be one more than the value of the free-running counter on the rising edge of the internal bus clock preceding the external transition. This delay is required for internal synchronization. Resolution is one count of the free-running counter, which is four internal bus clock cycles.

The free-running counter contents are transferred to the input capture register on each proper signal transition regardless of whether the input capture flag (ICF) is set or clear. The input capture register always contains the free-running counter value that corresponds to the most recent input capture.

After a read of the input capture register ($14) MSB, the counter transfer is inhibited until the LSB ($15) also is read. This characteristic causes the time used in the input capture software routine and its interaction with the main program to determine the minimum pulse period.

A read of the input capture register LSB ($15) does not inhibit the free-running counter transfer since they occur on opposite edges of the internal bus clock.

8.1.4 TIMER CONTROL REGISTER (TCR) $12

The TCR is a read/write register containing five control bits. Three bits enable interrupts associated with the timer status register flags ICF, OCF, and TOF.

	B7	B6	B5	B4	B3	B2	B1	B0	
$0012	ICIE	OC1IE	TOIE	0	0	0	IEDG	OLVL	TCR
RESET:	0	0	0	0	0	0	U	0	

READ: anytime

WRITE: anytime

ICIE Input Capture Interrupt Enable

1 - Interrupt enabled
0 - Interrupt disabled

OC1IE Output Compare 1 Interrupt Enable

1 - Interrupt enabled
0 - Interrupt disabled

TOIE Timer Overflow Interrupt Enable

1 - Interrupt enabled
0 - Interrupt disabled

BITS 2-4 Not used

Always read zero

IEDG Input Edge

Value of input edge determines which level transition on TCAP pin will trigger free-running counter transfer to the input capture register

Reset does not affect the IEDG bit (U = unaffected).

1 - Positive edge
0 - Negative edge

OLVL Output Level

Value of output level is clocked into output level register by the next successful output compare and will appear on the TCMP pin if DDRG3 also is set. This bit and the output level register are cleared by reset.

1 - High output
0 - Low output

8.1.5 TIMER STATUS REGISTER (TSR) $13

The TSR is a read-only register containing three status flag bits.

	B7	B6	B5	B4	B3	B2	B1	B0	
$0013	ICF	OC1F	TOF	0	0	0	0	0	TSR
RESET:	U	U	U	0	0	0	0	0	

READ: anytime

WRITE: no effect

ICF **Input Capture Flag**

1 - Flag set when selected polarity edge is sensed by input capture edge detector

0 - Flag cleared when TSR and input capture low register ($15) are accessed

OC1F **Output Compare 1 Flag**

1 - Flag set when output compare register contents match the free-running counter contents

0 - Flag cleared when TSR and output compare low register ($17) are accessed

TOF **Timer Overflow Flag**

1 - Flag set when free-running counter transition from $FFFF to $0000 occurs

0 - Flag cleared when TSR and counter low register ($19) are accessed

BITS 0-4 **Not used**

Always read zero

Accessing the timer status register satisfies the first condition required to clear status bits. The remaining step is to access the register corresponding to the status bit.

A problem can occur when using the timer overflow function and reading the free-running counter at random times to measure an elapsed time. Without incorporating the proper precautions into software, the timer overflow flag could be cleared unintentionally if:

1) The timer status register is read or written when TOF is set, and
2) The LSB of the free-running counter is read but not for the purpose of servicing the flag.

The counter alternate register at address $1A and $1B contains the same value as the free-running counter (at address $18 and $19); therefore, this alternate register can be read at any time without affecting the timer overflow flag in the timer status register.

8.1.6 TIMER DURING WAIT MODE

The CPU clock halts during the wait mode, but the timer remains active. If interrupts are enabled, a timer interrupt will cause the processor to exit the wait mode.

8.1.7 TIMER DURING STOP MODE

In stop mode, the timer stops counting and holds the last count value if stop is exited by an interrupt. If RESET is used, the counter is forced to $FFFC. During stop, if at least one valid input capture edge occurs at the TCAP pin, the input capture detect circuit is armed. This does not set any timer flags or wakeup the MCU, but when the MCU does wakeup, there is an active input capture flag and data from the first valid edge that occurred during stop mode. If RESET is used to exit stop mode, then no input capture flag or data remains, even if a valid input capture edge occurred.

8.2 TIMER 2

Timer 2 is an 8-bit event counter which has one compare register, event input pin (EVI), and event output pin (EVO). The event counter is clocked by the external clock (EXCLK) or prescaled system clock (CLK2) that is selected by the T2CLK bit in the TCR2 register. The EXCLK may be EVI direct or EVI gated by CLK2, which is selected by the IM2 bit at the EVI block. (Refer to the EVI description.)

Timer 2 may be used as a modulus clock divider with EVO pin, free-running counter (when compare register is $00), or periodic interrupt timer.

The timer counter 2 is an 8-bit up counter with preset input. The counter is preset to $01 by the CMP2 signal from the comparator or a CPU write to this counter (TCNT2), done while the system clock (PHI2) is low.

The CLK2 from the prescaler or the EXTCLK from the EVI block are selected as timer clock by the T2CLK bit in the TCR2 register. The CLK2 and the EXCLK are synchronized to the falling edge of system clock in the prescaler and the EVI blocks. The minimum pulse width of CLK2 is the same as the system clock, and the minimum pulse width of EXCLK (event mode) is one PHI2 cycle. When the EXCLK (event mode) is selected, 50% duty is not guaranteed.

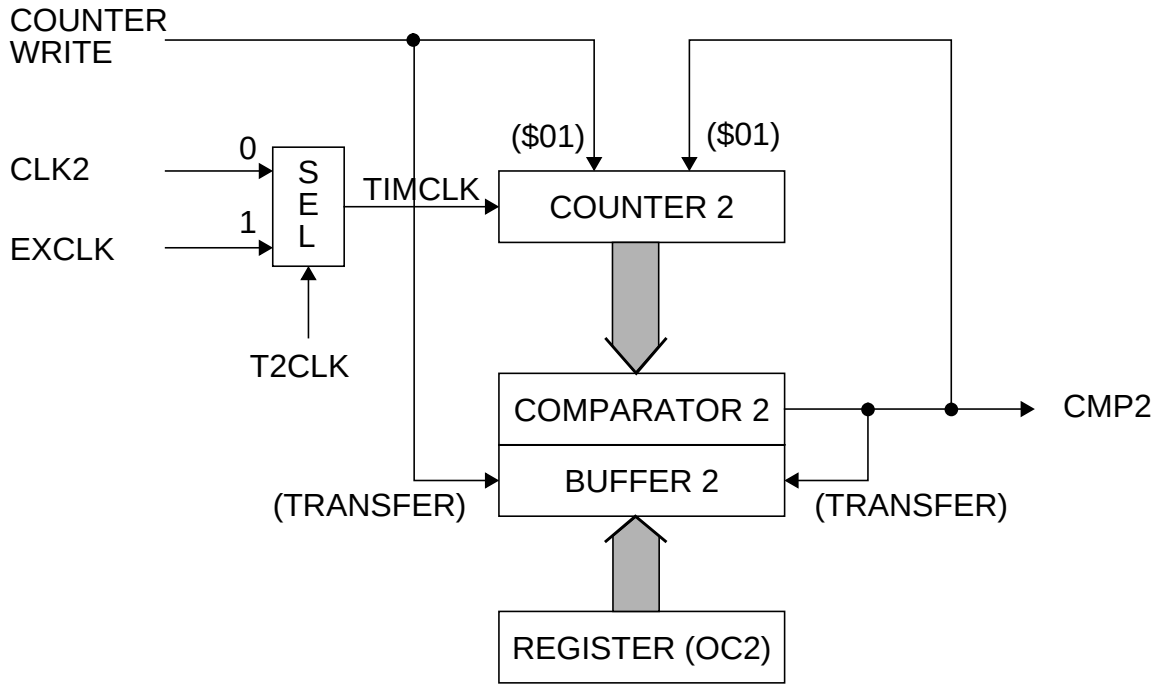


Figure 8-3: Timer 2 Block Diagram

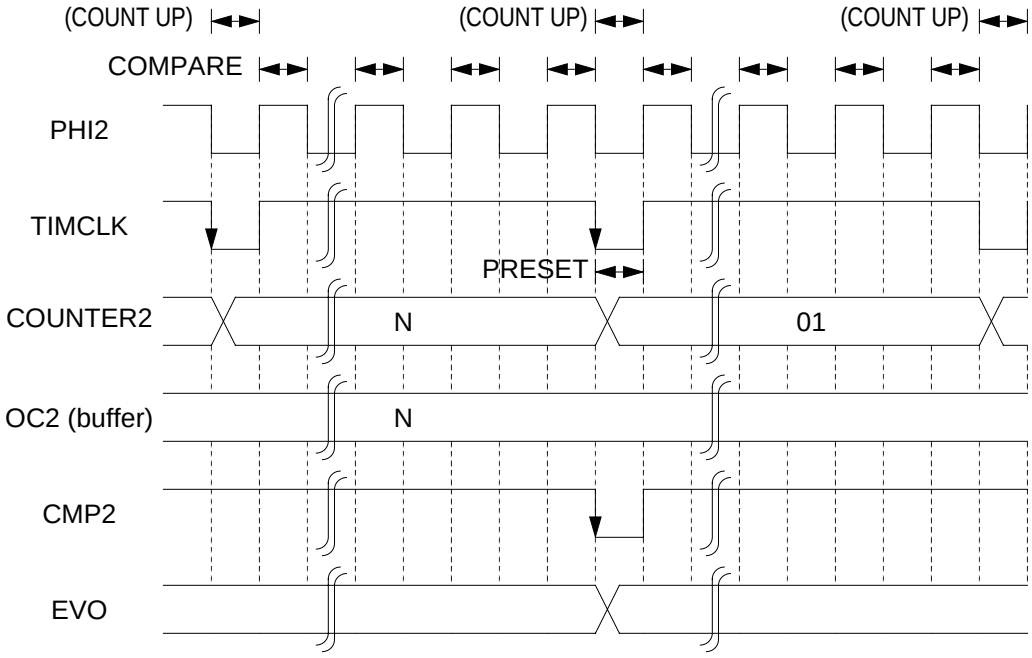
The counter is incremented by the falling edge of the timer clock and the period between two falling edges is defined as one timer cycle in the following description.

The compare register (OC2) is provided for the comparison with the timer counter. The OC2 data is transferred to the buffer register when the counter is preset by the CPU write or the compare output (CMP2). Actually, this buffer register is compared with the timer counter.

The comparison between the counter and the OC2 buffer register is done while the system clock is high in each bus cycle. If the counter matches with the OC2 buffer register, the comparator latches this result during the current timer cycle. When the next timer cycle begins, the comparator outputs CMP2 signal (if the compare match is detected during the previous timer cycle). This CMP2 is used in the counter preset, data transfer to the buffer register, setting OC2F in the TSR2, and the EVO block. The counter preset overrides the counter increment.

The OC2F bit may generate interrupt request if the OC2IE bit in the TCR2 is set to 1.

OC2=$2,3,4,...,FF,0



OC2=$

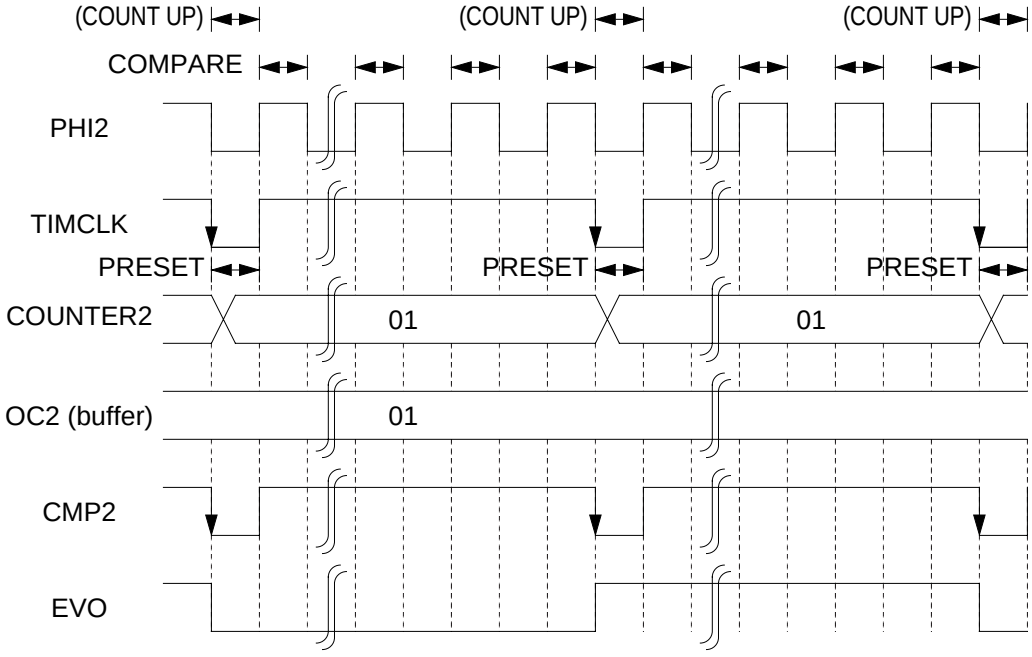


Figure 8-4: Timer 2 Timing for f(PHI2) > f(TIMCLK)

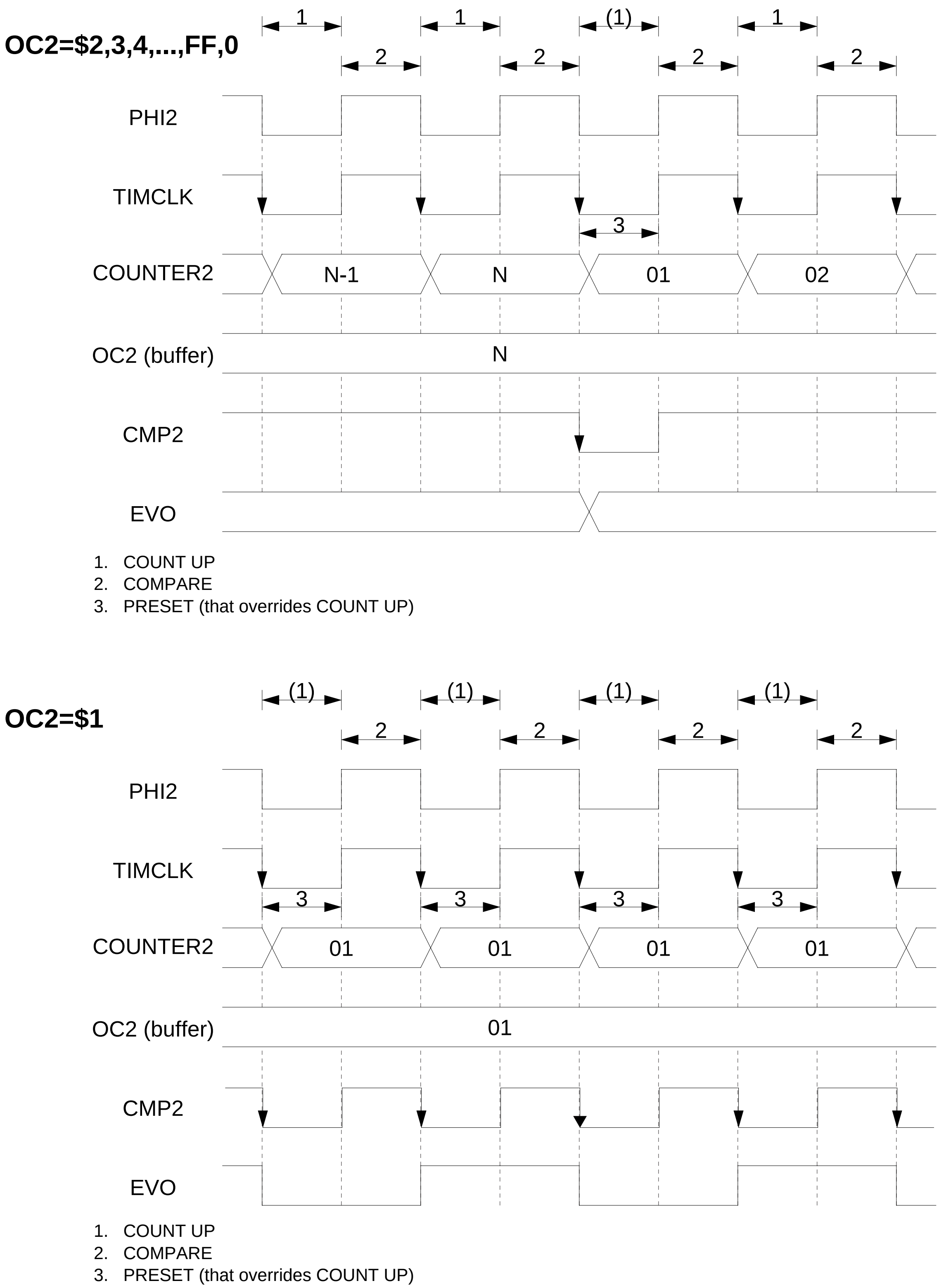


Figure 8-5: Timer 2 Timing for f(PHI2) = f(TIMCLK)

8.3 PRESCALER

The 8-bit prescaler in the timer system divides the system clock (PHI2) and provides the divided clock to timers and event input. The 3-bit prescaler provides divided clock to the PWM. See **Figure 8-6: Prescaler Block Diagram**.

CLK1 for the timer 1 is a fixed frequency clock (PHI2/4).

CLK2 for the timer 2 is selected by the T2R1 and T2R0 bits in the TBCR1, and this clock is used at the event input for the gate mode. The CLK2 transitions must be synchronous to the falling edge of PHI2.

CLK3 for the PWM is selected by the T3R1 and T3R0 bits in the TBCR1, and this clock is for the PWM counter. The CLK3 transitions must be synchronous to the falling edge of PHI2.

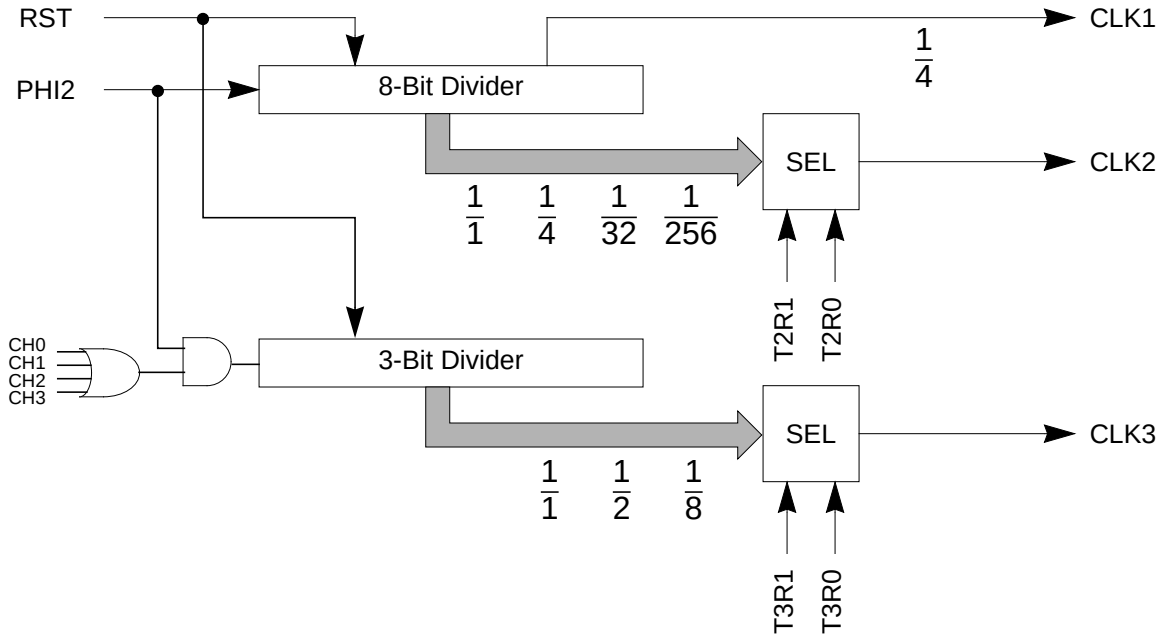


Figure 8-6: Prescaler Block Diagram

8.4 TIMER I/O PINS

Two input (TCAP and EVI) and two output (TCMP and EVO) pins are reserved for the timers.

8.4.1 TIMER INPUT 1 (TCAP)

This input pin is used for the input capture of timer 1. Active input edge (rising edge or falling edge) is selected by the IEDG bit in the TCR. Since the TCAP pin is shared with the PC3 I/O pin, changing the state of the DDRC3 or data register can cause an unwanted TCAP interrupt. This can be handled by clearing the ICIE bit before changing the configuration of PC3 and clearing any pending interrupts before enabling ICIE.

8.4.2 TIMER INPUT 2 (EVI)

The event input (EVI) is used as external clock input of the timer 2.

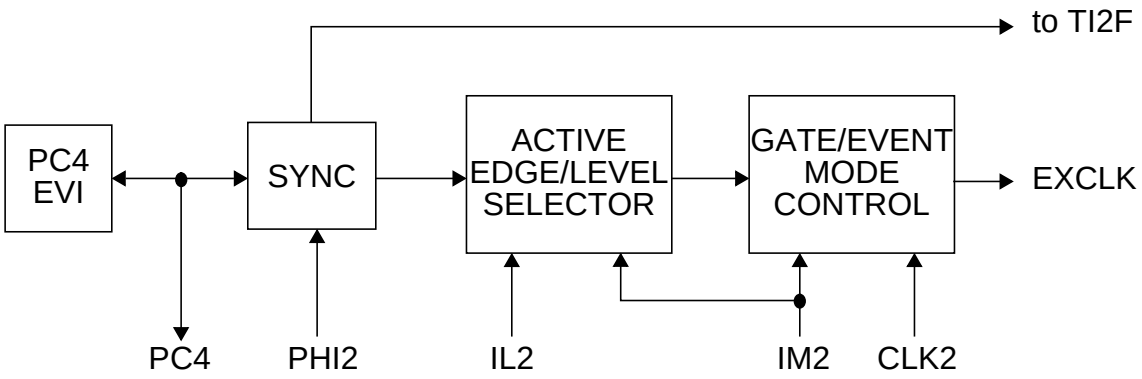


Figure 8-7: EVI Block Diagram

Since the external clock may be asynchronous to the internal clock, this input has a synchronizer which samples the external clock by the internal system clock. (The input transition synchronizes to the falling edge of PHI2. Therefore, the minimum pulse width for EVI should be larger than one system clock.)

The IM2 and IL2 bits in the TCR2 determine how this synchronized external clock is used. IM2 bit selects either the event mode or gated mode, and IL2 bit selects whether the level or edge is activated.

In the event mode (IM2 = 0), the external clock drives the timer 2 counter directly and the active edge at the EVI pin is selected by the IL2 bit. When the active edge is detected, the TI2F bit in the TSR2 is set.

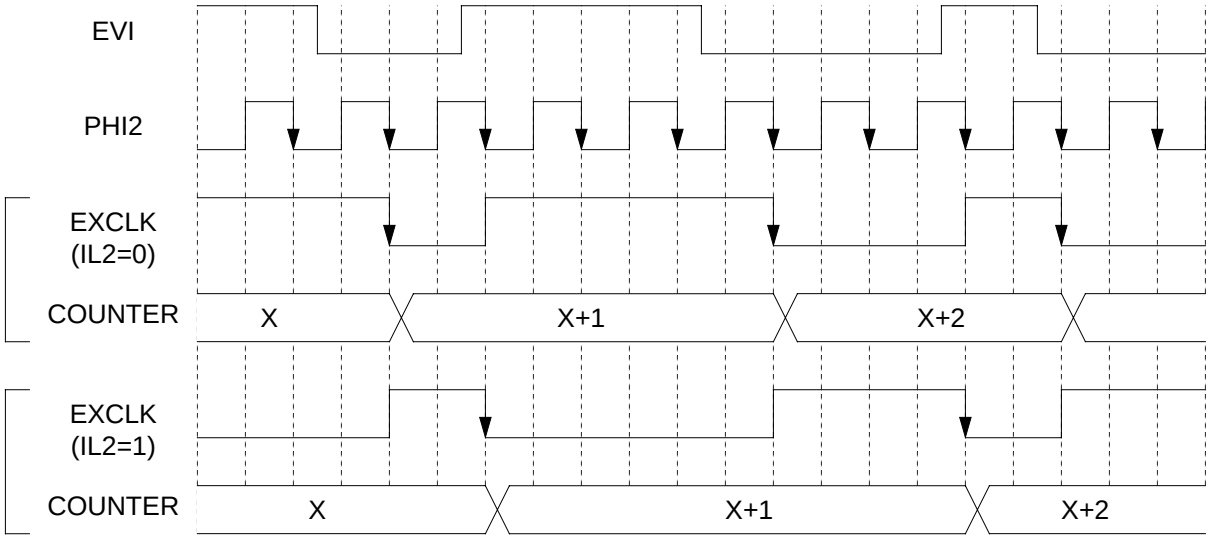
In the gated mode (IM2 = 1), the EVI input is gated by CLK2 from the prescaler and the gate output drives the timer 2 counter. IL2 bit selects the active level of the external input. When the transition from active level to inactive level is detected, the TI2F bit is set.

Changing the IM2 bit may cause an illegal count up of TCNT2. Therefore, the software must preset the TCNT2 after initializing IM2. Since the EVI pin is shared with the PC4 I/O pin, DDRC4 always should be cleared whenever EVI is used. EVI cannot be used if DDRC4 is high.

Table 8-1: EVI Mode Select

IM2	IL2	Action on Clock
0	0	EVI Falling Edge Increments Counter
0	1	EVI Rising Edge Increments Counter
1	0	Low Level on EVI Enables Counting
1	1	High Level on EVI Enables Counting

IM2=0 (Event Mode)



IM2=1 (Gate Mode)

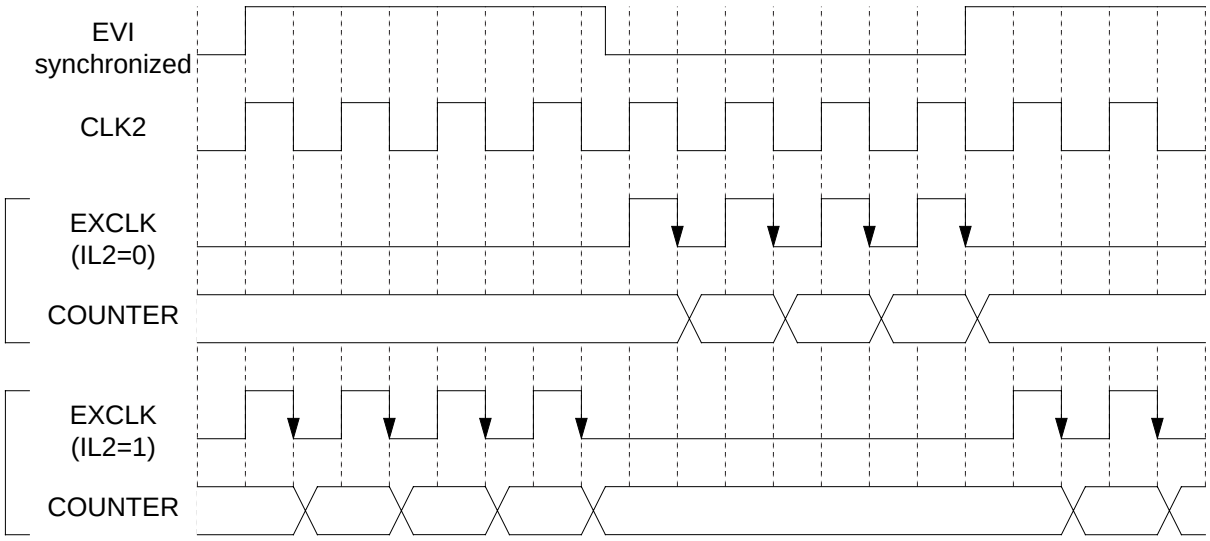


Figure 8-8: EVI Timing Examples

8.4.3 TIMER OUTPUT 1 (TCMP)

This output pin is used for the output compare of timer 1. Active output level (high or low level) is selected by the OLVL bit in the TCR.

8.4.4 TIMER OUTPUT 2 (EVO)

The EVO pin is the clock output pin of timer 2. The compare output from the timer 2 (CMP2) is divided in this block for 50% duty output signal. This 1/2 divider is initialized to the level of the OL2 bit when the timer counter 2 is written by the CPU (initialized).

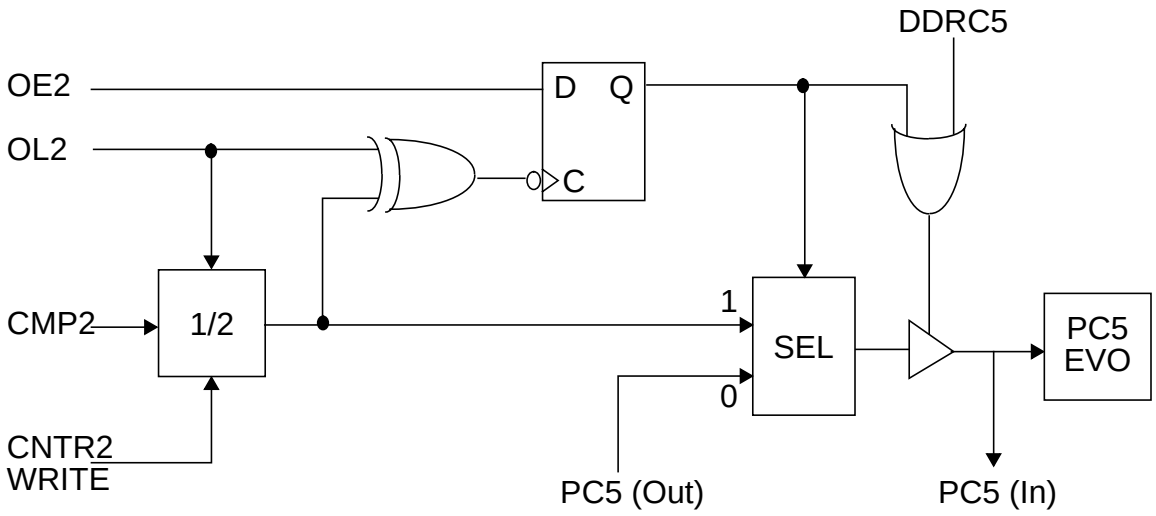


Figure 8-9: EVO Block Diagram

When the OE2 bit in the timer control register 2 (TCR2) is set, the EVO output is activated and when OE2 is cleared EVO is deactivated.

The output buffer at the EVO/PC5 pin is enabled when the DDRC5 bit is set or the synchronized output enable is high (clock on). If DDRC5 bit is set to one, the pin state during the idling condition (clock off) is decided by the PC5 data latch. If DDRC5 is cleared, the pin becomes high impedance during clock off.

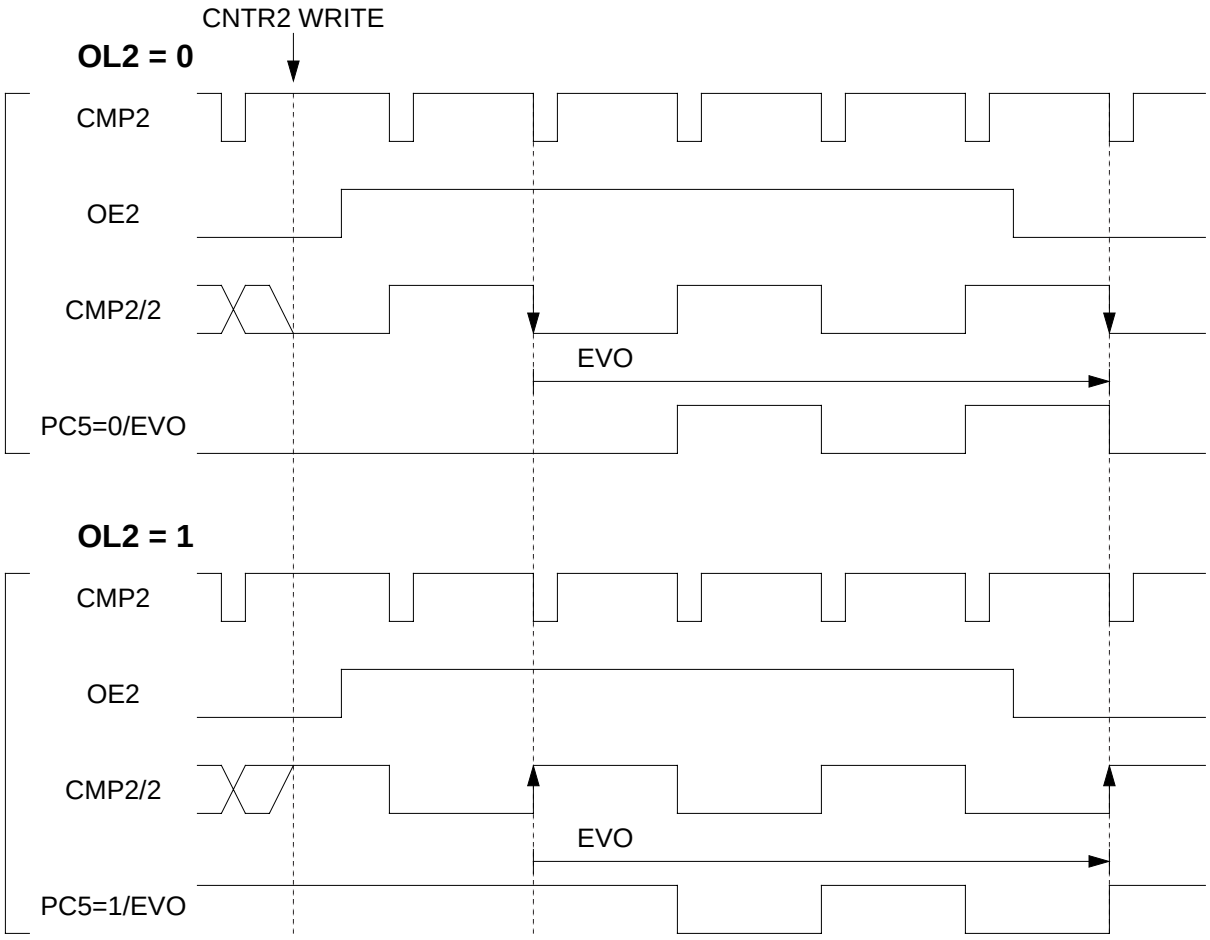


Figure 8-10: EVO Timing Example

8.5 TIMER REGISTERS

The timers have 14 registers and four rate select bits are added into the TBCR1 register. The 10 registers of timer 1 are compatible with the registers of MC68HC05C4. Refer to the C4 specifications for details on these registers.

8.5.1 TIMER CONTROL REGISTER 2 (TCR2)

	B7	B6	B5	B4	B3	B2	B1	B0	
$001C	TI2IE	OC2IE	0	T2CLK	IM2	IL2	OE2	OL2	TCR2
RESET:	0	0	0	0	0	0	0	0	

READ: anytime

WRITE: anytime

TI2IE Timer Input 2 Interrupt Enable

TI2IE bit enables timer input 2 (EVI) interrupt when TI2F is set. This bit is cleared at reset.

0 - Timer input 2 interrupt is disabled
1 - Timer input 2 interrupt is enabled

OC2IE Compare 2 Interrupt Enable

OC2IE bit enables compare 2 (CMP2) interrupt when compare match is detected (OC2F is set). This bit is cleared at reset.

0 - Compare 2 interrupt is disabled
1 - Compare 2 interrupt is enabled

BIT 5 Reserved

This bit is not used and always read as zero.

T2CLK Timer 2 Clock Select

The T2CLK bit selects clock source for the timer counter 2. This bit is cleared at reset.

0 - CLK2 from prescaler is selected
1 - EXCLK from EVI input block is selected

IM2 Timer Input 2 Mode Select

The IM2 bit selects whether EVI input is gated by CLK2 or not gated by CLK2. This bit is cleared at reset.

0 - EVI is not gated by CLK2 (event mode)
1 - EVI is gated by CLK2 (gated mode)

IL2 **Timer Input 2 active edge (Level) select**

The IL2 bit selects the active edge of EVI to increment counter for the event mode (IM2 = 0), or gate enable level of EVI for the gate mode (IM2 = 1). This bit is cleared at reset.

0 - Falling edge is selected (event mode)
Low level enables counting (gate mode)
1 - Rising edge is selected (event mode)
High level enables counting (gated mode)

IM2	IL2	Action on Clock
0	0	EVI Falling Edge Increments Counter
0	1	EVI Rising Edge Increments Counter
1	0	Low Level on EVI Enables Counting
1	1	High Level on EVI Enables Counting

OE2 **Timer Output 2 (EVO) Output Enable**

The OE2 bit enables EVO output on PC5 pin. When this bit is changed, the control of the pin is delayed (synchronized) until the next active edge of EVO is selected by OL2 bit occurs. This bit is cleared at reset.

0 - EVO output is disabled
1 - EVO output is enabled

OL2 **Timer Output 2 Edge select for synchronization**

The OL2 bit selects which edge of EVO clock should be synchronized by the OE2 bit control. The OL2 bit also decides the initial value of the CMP2 divider, when counter 2 is written by the CPU. This bit is cleared at reset.

0 - The falling edge of EVO switches EVO output and PC5 if the OE2 bit has been changed.
1 - The rising edge of EVO switches EVO output and PC5 if the OE2 bit has been changed.

8.5.2 TIMER STATUS REGISTER 2 (TSR2)

	B7	B6	B5	B4	B3	B2	B1	B0	
$001D	TI2F	OC2F	0	0	RTI2F	ROC2F	0	0	TSR2
RESET:	0	0	0	0	0	0	0	0	

READ: anytime
(Bits 3-2 are write-only bits and always read as zero.)

WRITE: anytime
(Bits 7-6 are read-only bits and write has no effect.)

TI2F Timer Input 2 (EVI) Interrupt Flag

In event mode, the event edge sets TI2F and in gated time accumulation mode the trailing edge of the gate signal at the EVI input pin sets TI2F. When TI2IE bit and this bit are set, interrupt is generated. This bit is a read-only bit and writes have no effect. The TI2F is cleared by writing a one to the RTI2F bit or by reset.

OC2F Compare 2 Interrupt Flag

The OC2F bit is set when a compare match is detected between counter 2 and OC2 register. If the OC2IE bit and this bit are set, interrupt is generated. This bit is a read-only bit and writes have no effect. The OC2F is cleared by writing a one to the ROC2F bit or by reset.

BITS 5-4 Reserved

These bits are not used and always read as zero.

RTI2F Reset Timer Input 2 Flag

The RTI2F bit is a write-only bit and always read as zero. Writing one to this bit clears TI2F bit and writing a zero to this bit has no effect.

ROC2F Reset Output Compare 2 Flag

The ROC2F bit is a write-only bit and always read as zero. Writing one to this bit clears the OC2F bit and writing a zero to this bit has no effect.

BITS 1-0 Reserved

These bits are not used and always read as zero.

8.5.3 OUTPUT COMPARE REGISTER 2 (OC2)

	B7	B6	B5	B4	B3	B2	B1	B0	
$001E									OC2
RESET:	1	1	1	1	1	1	1	1	

READ: anytime

WRITE: anytime

The data in the OC2 register is transferred to the buffer register when the CPU writes to the TCNT2, when the CMP2 presets the TCNT2, or when the system resets.

When the OC2 buffer register matches the TCNT2 register, the OC2F bit in the TSR2 register is set and TCNT2 is preset to $01. OC2 is preset to $FF on reset.

8.5.4 TIMER COUNTER 2 (TCNT2)

	B7	B6	B5	B4	B3	B2	B1	B0	
$001F									TCNT2
RESET:	0	0	0	0	0	0	0	1	

READ: anytime

WRITE: anytime (TCNT2 becomes $01 by any write data)

The timer counter 2 (TCNT2) is incremented by the falling edge of the timer clock (which is synchronized and has the same timing as the falling edge of PHI2).

The TCNT2 register is compared with the OC2 buffer register and initialized to $01 if it matches.

This counter also is initialized to $01 by reset or any CPU write to this register. The CPU read of this counter should be done while PHI2 is high and data may be latched by the local or main data bus while PHI2 is low.

8.5.5 TIMER BASE CONTROL REGISTER 1 (TBCR1)

	B7	B6	B5	B4	B3	B2	B1	B0	
$0010	TBCLK	0	0	0	T3R1	T3R0	T2R1	T2R0	TBCR1
RESET:	0	0	0	0	0	0	0	0	

READ: anytime

WRITE: anytime (only one-time write is allowed on bit 7 after reset)

TBCLK **Time Base Clock**

The TBCLK bit selects time base clock source. This bit is cleared at reset. After reset, write to this bit is allowed only once.

0 - XOSC clock is selected
1 - OSC clock divided by 128 is selected

BITS 6-4 **Reserved**

These bits are not used and always read as 0.

T3R1/0 **Prescale Rate and Clock Select Bits for PWM**

The T3R1 and T3R0 bits select the prescale rate of CLK3 or CMP2 from timer 2 for the PWM. These bits are cleared by reset.

T3R1	T3R0	PWM CLOCK	
0	0	E	(CLK3)
0	1	E/2	(CLK3)
1	0	E/8	(CLK3)
1	1	TIMCLK*/N (N=1....256)	(CMP2)

* TIMCLK = CLK2 or EXCLK

NOTE: While bits T3R1 and T3R0 may be written any time, if the selection is changed while a PWM signal is being generated, a truncated or stretched pulse may occur during the transition.

T2R1/0 Prescale Rate Select Bits for Timer 2

The T2R1 and T2R0 bits select the prescale rate of CLK2 for timer 2 and timer input 2. These bits are cleared by reset.

T2R1	T2R0	System Clock Divided by
0	0	1
0	1	4
1	0	32
1	1	256

SECTION 9 PULSE WIDTH MODULATOR

9.1 GENERAL

The 4-channel, 8-bit pulse width modulator (PWM) system works in conjunction with an 8-bit up counter with preset input. This timer behaves similarly to timer 2 except for the way it transfers the data and presets the counter. See **Figure 9-1: PWM System Block Diagram**.

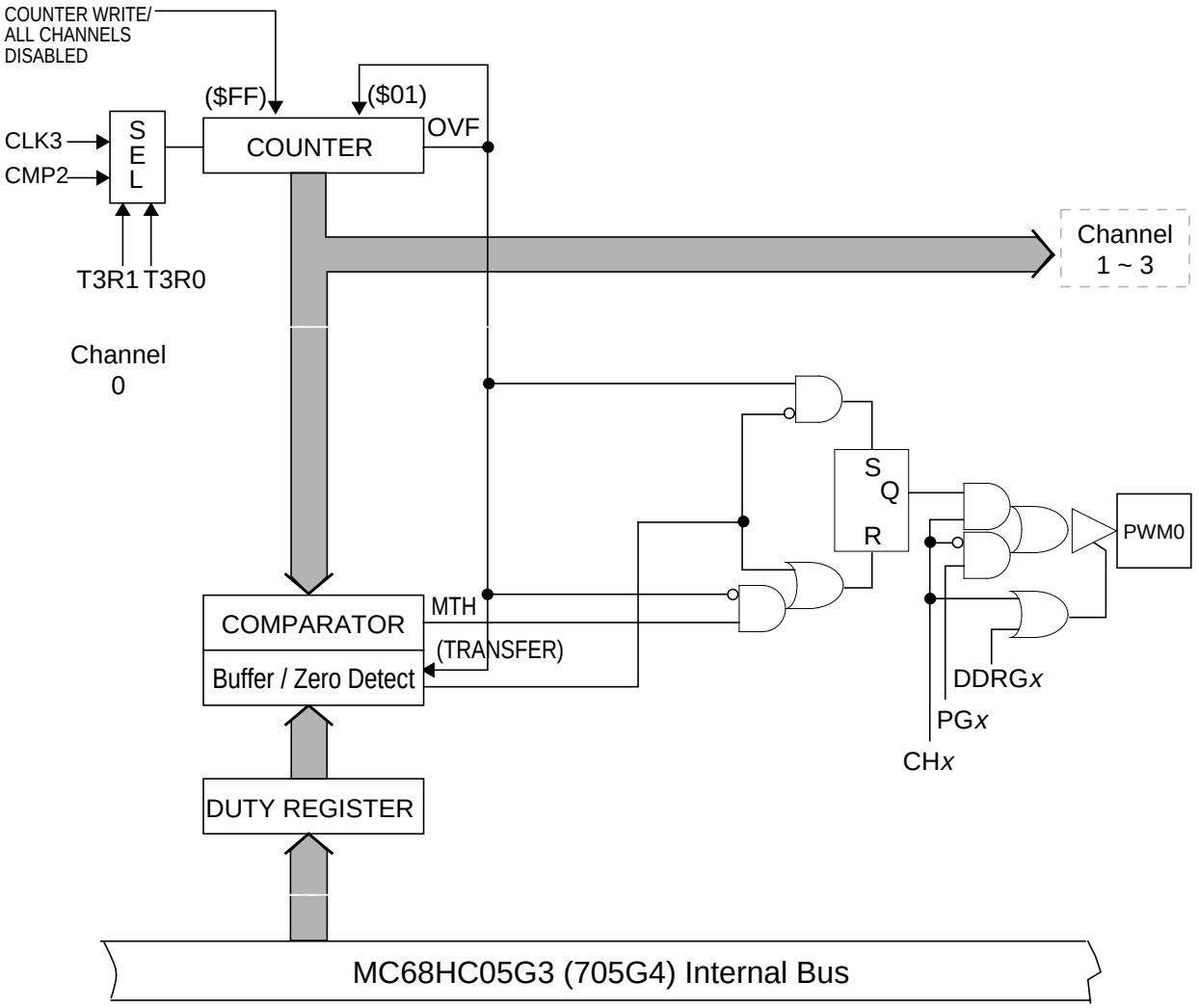


Figure 9-1: PWM System Block Diagram

A flexible clock selection scheme allows two different clocks to be used with the counter. CLK3 from the prescaler or the modulus clock (CMP2) from timer 2 can be selected as the clock source by the T3R 1/0 bits in the TBCR1 register. This gives a programmable period of 255 x (1/T), where T can be E, E/2, or E/8 (E = bus frequency) when CLK3 is selected

or TIMCLK is divided by any positive integer up to 256 when CMP2 is selected. Obviously, CMP2 should be selected only if timer 2 is not being used or if it is generating a desired frequency that could be shared. Refer to 8.2 Timer 2 for more information on using CMP2. The following table shows the PWM clock selections.

Table 9-1: PWM Clock Selection

T3R1	T3R0	PWM CLOCK
0	0	E (CLK3)
0	1	E/2 (CLK3)
1	0	E/8 (CLK3)
1	1	TIMCLK*/N (N=1....256) (CMP2)

* TIMCLK = CLK2 or EXCLK

NOTE: While bits T3R1 and T3R0 may be written any time, if the selection is changed while a PWM signal is being generated, a truncated or stretched pulse may occur during the transition. To prevent this from happening, it is recommended that all PWM channels be disabled or the counter be forced to $FF when changing clock selections.

CLK3 from the prescaler is activated by enabling the PWM channel(s). This is done to ensure that the moment the first PWM channel(s) is enabled, the counter can start incrementing without any clock delays. This does not apply to CMP2, since CMP2 is controlled by timer 2.

The counter is incremented by the falling edge of the timer clock and is either preset to $01 by the overflow (OVF) from the counter, $FF by disabling all PWM channels, or writing to this counter (PWMCNT) while the system clock (PHI2) is low.

Since only one counter is shared by all the channels, only the first PWM signal output(s) can be synchronized to the starting edge of the CLK3 clock when the channel(s) is enabled. This first PWM signal output(s) will initiate with a complete PWM period. Any channel enabled after the starting edge of the CLK3 clock will generate a truncated pulse during the initial period.

Each channel has its own 8-bit duty register which is double buffered. When a channel is active (enable bit is high), writes to the duty register are buffered until the counter rolls over. At this time the new duty takes effect. In this way, the output of the PWM always will be either the old duty waveform or the new duty waveform, not some variation in between.

A change in duty can be forced into effect immediately by writing the new value to the duty register and then writing any value to the counter. This causes the counter to reset to $FF and the newly latched duty value to be transferred to the buffer. In addition, since the counter is readable, it is possible to know where the count is with respect to the duty value and software can be used to make the adjustments.

A brief operational description of a PWM channel: An 8-bit counter runs at the rate of the selected clock source, as described earlier. When all channels are disabled, the counter is preset to $FF. Once the channel(s) is enabled, the counter begins incrementing. When this counter overflows, three things happen: the counter presets to $01, a flip-flop is set causing the PWM output to go high, and the latched value in the duty register is transferred to the buffer. A match (MTH) between the counter and the duty register buffer resets the flip-flop, thus giving a low output. A value $00 written into the duty register buffer triggers the zero detector to hold the reset on the flip-flop.

9.2 PWM CONTROL REGISTER (PWMCR)

Each channel of the PWM is enabled by a bit in the PWMCR register.

	B7	B6	B5	B4	B3	B2	B1	B0	
$0034	0	0	0	0	CH3	CH2	CH1	CH0	PWMCR
RESET:	0	0	0	0	0	0	0	0	

Figure 9-2: PWM Control Registers

Each PWM output pin is shared with a general programmable port I/O bit. When the PWM output control bit (CH*x*) is set to one, the associated port G line will function as a PWM output regardless of the state of the associated DDRG bit. This does not change the state of the DDR bit, and when CH*x* is disabled the DDRG*x* bit again controls the I/O state. CH*x* is cleared on reset to prevent erroneous output.

9.3 PWM DUTY REGISTER (PWMDR*x*)

The PWM has four duty registers associated with it: $36 - $39, PWMDR0 - PWMDR3. Reads of this register return the most recent written value.

	B7	B6	B5	B4	B3	B2	B1	B0	
$36-39									PWMDR*x*
RESET:	0	0	0	0	0	0	0	0	

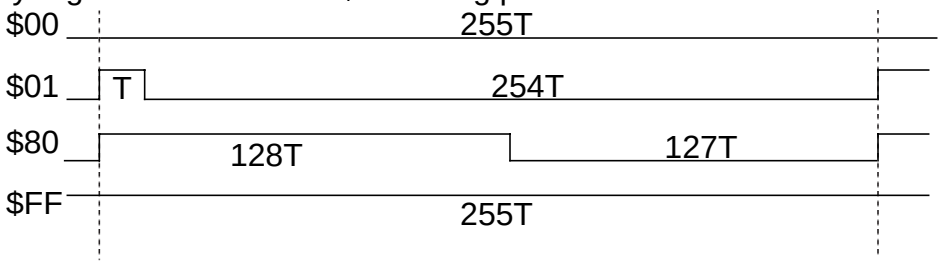
Figure 9-3: PWM Duty Registers

Each output is a pulse width modulated signal whose duty cycle varies according to the value set into its duty register. The duty cycle is expressed with eight bits of resolution. The signal can be used directly as a PWM signal, or it may be filtered to obtain an average value for a general-purpose analog output.

The repetition rate is 255 times the programmable timer clock overflow rate. (For example, the repetition rate for a 4.00 MHz crystal (2 MHz internal clock) is 7843 Hz.) A value of $00 loaded into the duty register results in a continuous low output on the corresponding PWM output pin. A value of $7F or $80 results in approximately 50% duty cycle output, and so on, to the maximum value, $FF, which corresponds to an output which is at 1 for 255/255 of the cycle. If the register PWMDR*x* is written while the channel is enabled, the new value will be picked up by the PWM converters only at the end of a complete conversion cycle.

This results in a monotonic change of the DC component of the output without overshoots or vicious starts. (A vicious start is an output which gives totally erroneous PWM during the initial period following an update of the PWM register.) This feature is achieved by double buffering of the PWM registers.

All PWM duty registers are reset to $00 during power-on or external reset.



1/255T = 3921 Hz, and T = 1/(255X3921) ≅ 1.0 µS, so T = 2 CPU clocks

Figure 9-4: PWM Waveform Examples (E = 2MHz; CLK = E/2)

9.4 PWM COUNTER (PWMCNT)

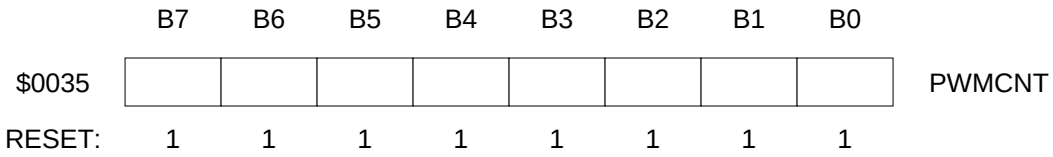


Figure 9-5: PWM Counter

The PWM counter may be read any time without affecting the count or the operation of the PWM channel. The PWM counter (PWMCNT) is incremented by the falling edge of the PWM timer clock (which is synchronized and has the same timing as the falling edge of PHI2).

The PWMCNT register is initialized to $01 if it overflows. However, this counter is initialized to $FF when this register is written by the CPU or all channels are disabled. Writes to the counter while the channel(s) is enabled (counting) may cause a truncated PWM period.

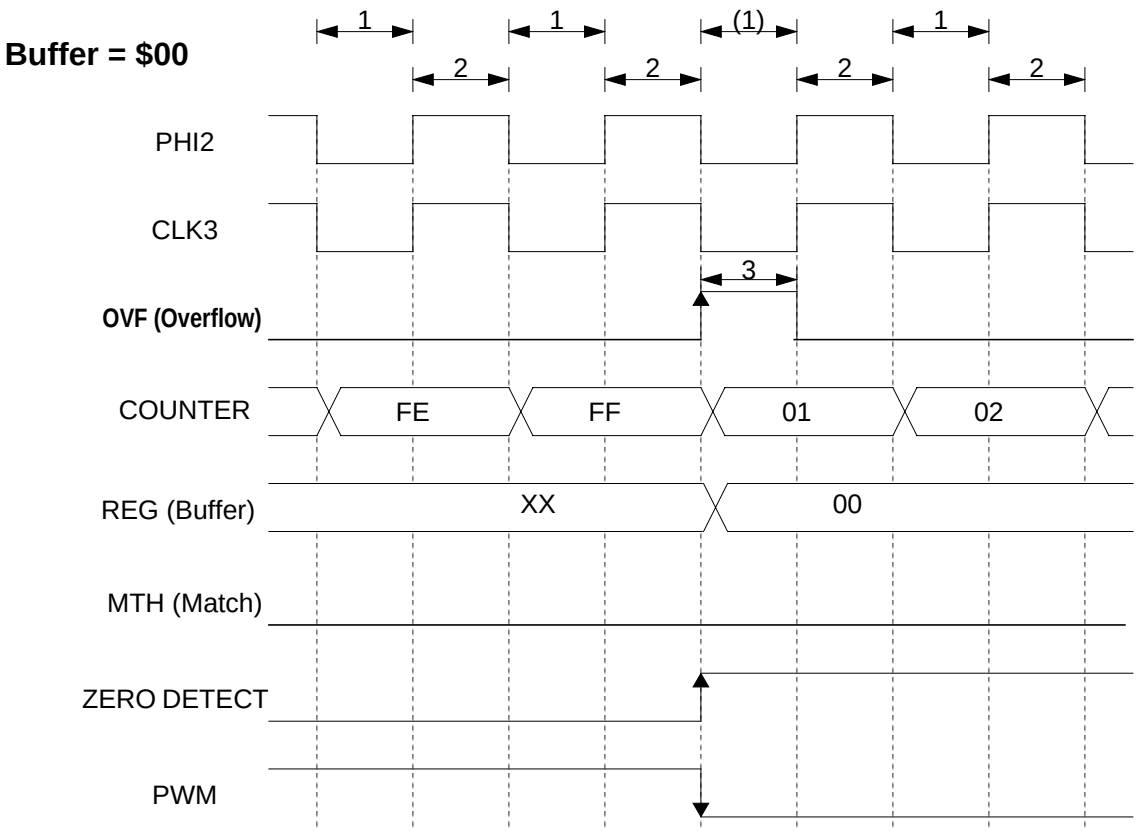
When the channel(s) is enabled (CH*x* written from zero to one), the PWM counter starts incrementing using whichever clock it has selected.

9.5 PWM DURING WAIT MODE

The PWM continues normal operation during wait mode. To decrease power consumption during WAIT, it is recommended that the CH3-0 bits in the PWMCR register be cleared if the PWM is not being used.

9.6 PWM DURING STOP MODE

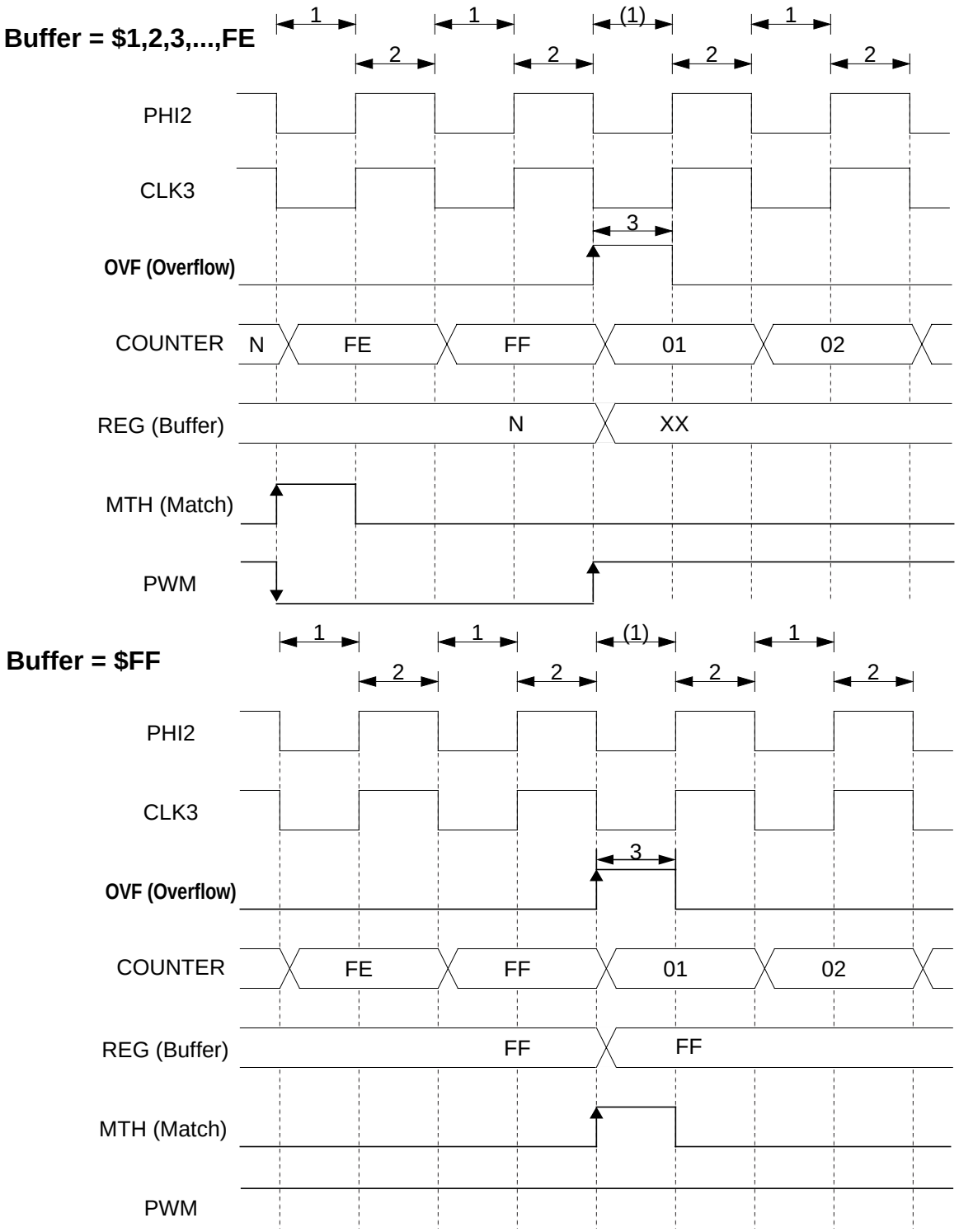
In stop mode the system clock is stopped causing the PWM to cease function. Any signal in process is suspended in whatever phase the signal happens to be in. When the clock begins oscillation upon leaving stop mode, the PWM will resume where it left off.



NOTE: Zero detect overrides the OVF signal.

1. COUNT UP
2. COMPARE
3. RESET (that overrides COUNT UP)

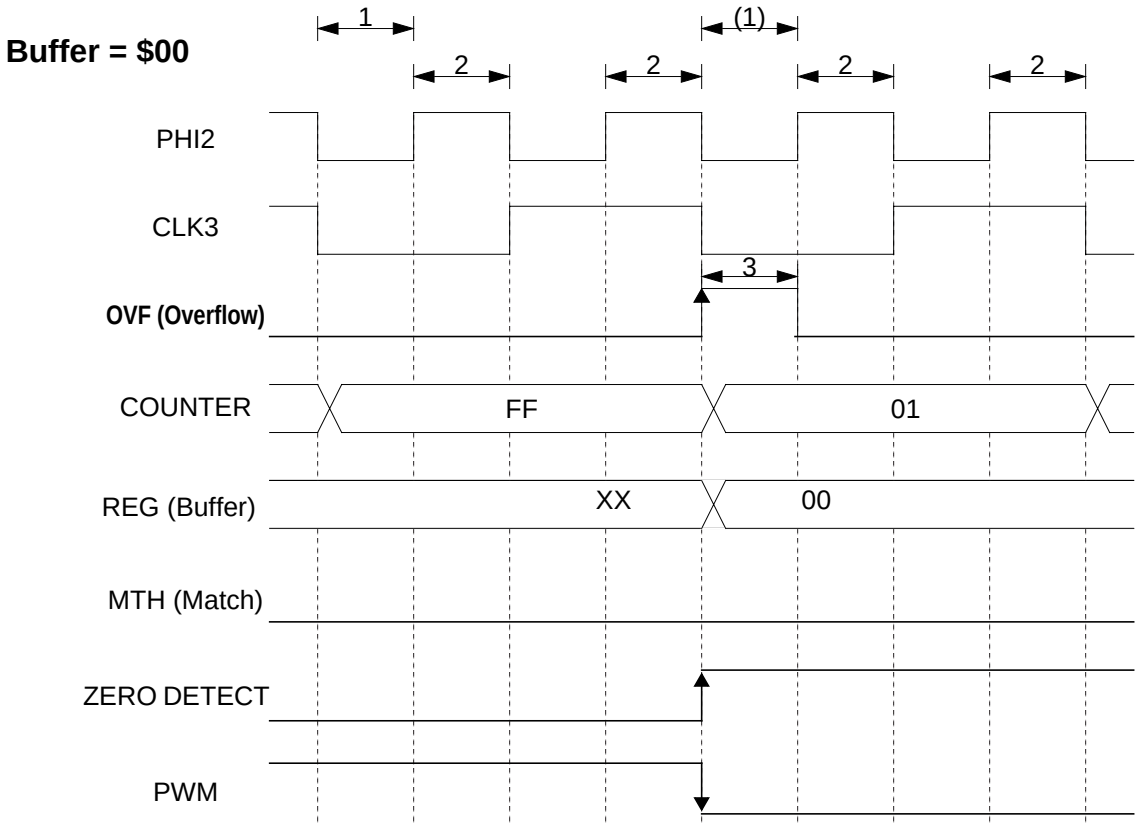
Figure 9-6: PWM Timing for f(CLK3) = f(PHI2)



NOTE: OVF overrides the MTH signal.

1. COUNT UP
2. COMPARE
3. PRESET (that overrides COUNT UP)

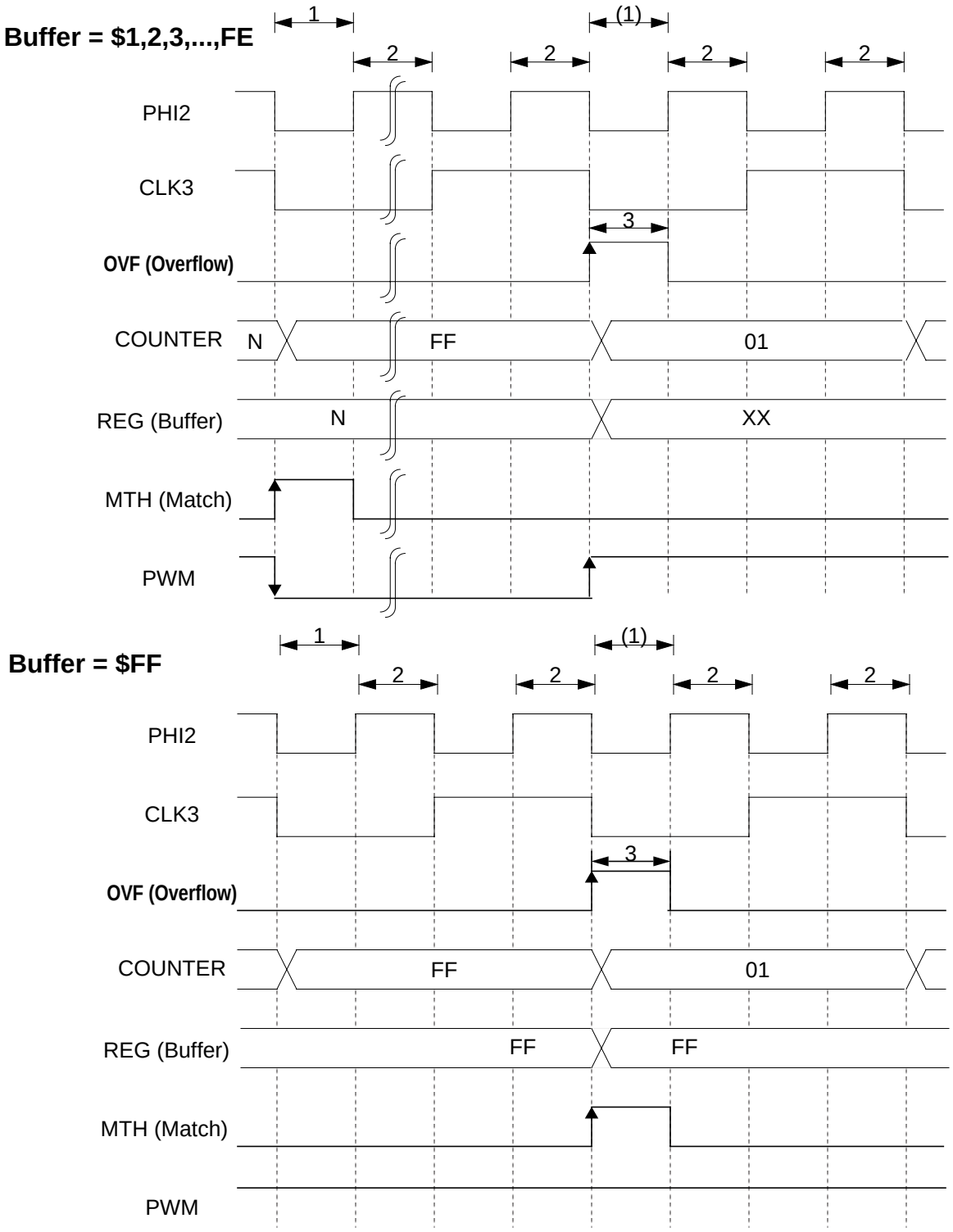
Figure 9-7: PWM Timing for f(CLK3) = f(PHI2)



NOTE: Zero detect overrides the OVF signal.

1. COUNT UP
2. COMPARE
3. PRESET (that overrides COUNT UP)

Figure 9-8: PWM Timing for f(CLK3) < f(PHI2)



NOTE: OVF overrides the MTH signal.

1. COUNT UP
2. COMPARE
3. PRESET (that overrides COUNT UP)

Figure 9-9: PWM Timing for f(CLK3) < f(PHI2

SECTION 10 A/D CONVERTER

The MC68HC05G3 (705G4) includes an 8-channel, 8-bit, multiplexed input, successive approximation A/D converter, with eight of the inputs available on external pins and four additional internal channels.

10.1 ANALOG SECTION

10.1.1 RATIOMETRIC CONVERSION

The A/D is ratiometric, with two dedicated pins supplying the reference voltages (V_{REFH} and V_{REFL}). An input voltage equal to V_{REFH} converts to $FF (full scale) and an input voltage equal to V_{REFL} converts to $00. An input voltage greater than V_{REFH} will convert to $FF with no overflow indication. For ratiometric conversions, the source of each analog input should use V_{REFH} as the supply voltage and be referenced to V_{REFL}.

10.1.2 V_{REFH} and V_{REFL}

The reference supply for the converter uses two dedicated pins rather than being driven by the system power supply lines because the voltage drops in the bonding wires of those heavily loaded pins would degrade the accuracy of the A/D conversion. V_{REFH} and V_{REFL} can be any voltage between V_{DD} and V_{SS} as long as $V_{REFH} > V_{REFL}$. However, the accuracy of conversions is tested and guaranteed only for $V_{REFH} = V_{DD}$ and $V_{REFL} = V_{SS}$.

10.1.3 ACCURACY AND PRECISION

The 8-bit conversions shall be accurate to within $\pm 1^1/_2$ LSB, including quantization at $V_{REFH} = V_{DD} = 5V$ and $V_{REFL} = V_{SS} = 0V$.

10.2 CONVERSION PROCESS

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 compared successively to the selected analog input which was sampled at the beginning of the conversion time. The conversion process is monotonic and has no missing codes.

10.3 DIGITAL SECTION

10.3.1 CONVERSION TIMES

Each channel of conversion takes 32 clock cycles, which must be at a frequency equal to or greater than 1 MHz.

10.3.2 MULTI-CHANNEL OPERATION

A multiplexer allows the single A/D converter to select one of eight analog signals. The eight pins of port F are input signals to the multiplexer.

10.4 A/D STATUS AND CONTROL REGISTER (ADSCR) $3B

The following paragraphs describe the function of the A/D status and control register.

	B7	B6	B5	B4	B3	B2	B1	B0	
$003B	COCO	ADRC	ADON	0	CH3	CH2	CH1	CH0	ADSCR
RESET:	0	0	0	0	0	0	0	0	

Figure 10-1: A/D Status and Control Register

10.4.1 COCO - CONVERSIONS COMPLETE

This read-only status bit is set when a conversion is completed, indicating that the A/D data register contains valid results. This bit is cleared whenever the A/D status and control register is written and a new conversion automatically started, or whenever the A/D data register is read. Once a conversion has been started by writing to the A/D status and control register, conversions of the selected channel will continue every 32 cycles until the A/D status and control register is written again. In this continuous conversion mode, the A/D data register will be filled with new data and the COCO bit will be set every 32 cycles. Data from the previous conversion will be overwritten regardless of the state of the COCO bit prior to writing.

10.4.2 ADRC - RC OSCILLATOR ON

When ADRC is set, the A/D section runs on the internal RC oscillator instead of the CPU clock. The RC oscillator requires a time, t_{ADRC} = 5µs, to stabilize and results can be inaccurate during this time. If the CPU clock is running below 1 MHz (using OSC, not XOSC), the RC oscillator must be used. When ADRC is cleared, the A/D uses the CPU clock.

When the RC oscillator is being used as the conversion clock, three limitations apply:

1. The conversion complete flag (COCO) must be used to determine when a conversion sequence has been completed, due to the frequency tolerance of the RC oscillator and its asynchronism with regard to the MCU E clock.
2. The conversion process runs at the nominal 1.5 MHz rate @ 5 V but 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 cycle.
3. If the system clock is running faster than the RC oscillator, the RC oscillator should be turned off, and the system clock used as the conversion clock.

10.4.3 ADON - A/D ON

When the A/D is turned on (ADON = 1), it requires a time, t_{ADON}= 100µs, for the current sources to stabilize, and results can be inaccurate during this time. This bit turns on the charge pump. Clearing ADON while the ADRC is set will disable the RC oscillator to save power. When ADON is cleared, the A/D converter is completely turned off with no current leakage.

10.4.4 CH3:CH0 - CHANNEL SELECT BITS

CH3, CH2, CH1, and CH0 form a 4-bit field which is used to select one of eight A/D channels. Channels 0 through 7 correspond to port F input pins on the MCU. Channels C through F are used for internal reference points. In user mode, channel F is reserved and converts to $00. The following table shows the signals selected by the channel select field.

Using a port F pin as both an analog and digital input simultaneously is prohibited to prevent excess power dissipation. When the A/D is enabled (ADON = 1) and one of the channels 0 through 7 is selected, the corresponding port F pin will appear as a logic zero to a digital read. The remaining port F pins will read normally. To digitally read all eight port F pins simultaneously, the A/D must be disabled (ADON = 0) or one of channels 8 through B must be selected.

Table 10-1: A/D Channel Assignments

CHANNEL	SIGNAL
0	AD0 PORT F BIT 0
1	AD1 PORT F BIT 1
2	AD2 PORT F BIT 2
3	AD3 PORT F BIT 3
4	AD4 PORT F BIT 4
5	AD5 PORT F BIT 5
6	AD6 PORT F BIT 6
7	AD7 PORT F BIT 7
8 ~ B	RESERVED
C	V_{REFH}
D	$(V_{REFH} + V_{REFL})/2$
E	V_{REFL}
F	FACTORY TEST

10.5 A/D DATA REGISTER ($3A)

One 8-bit result register is provided. This register is updated each time COCO is set.

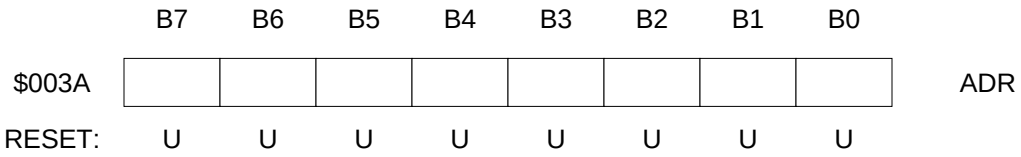


Figure 10-2: A/D Data Register

10.6 A/D DURING WAIT MODE

The A/D continues normal operation during wait mode. To decrease power consumption during WAIT, it is recommended that both the ADON and ADRC bits in the A/D status and control register be cleared if the A/D converter is not being used. If the A/D converter is in use and the system clock rate is above 1.0 MHz, it is recommended that the ADRC bit be cleared.

10.7 A/D DURING STOP MODE

In stop mode the comparator and charge pump are turned off and the A/D ceases to function. Any pending conversion is aborted. When the clocks begin oscillation upon leaving the stop mode, a finite amount of time passes before the A/D circuits stabilize enough to provide conversions to the specified accuracy. Normally, the delays built into the MC68HC05G3 when coming out of stop mode are sufficient for this purpose. Therefore, no explicit delays need to be built into the software.

SECTION 11 ELECTRICAL SPECIFICATIONS

11.1 MAXIMUM RATINGS

(Voltages referenced to V_{SS})

Rating	Symbol	Value	Unit
Supply Voltage	V_{DD}	-0.3 to +7.0	V
Input Voltage	V_{in}	V_{SS} - 0.3 to V_{DD} + 0.3	V
Bootloader Mode ($\overline{IRQ1}$, $\overline{IRQ2}$ Pins Only)	V_{TST}	V_{SS} - 0.3 to $2 \times V_{DD}$ + 0.3	V
Current Drain Per Pin Excluding V_{DD} and V_{SS}	I	25	mA
Operating Temperature Range MC68HC05G3 (Standard)	T_A	T_L to T_H -40 to +85	°C
Storage Temperature Range	T_{stg}	-65 to +150	°C

This device contains circuitry to protect the inputs against damage due to high static voltages or electric fields. However, it is advised that normal precautions be taken to avoid application of any voltage higher than maximum-rated voltages to this high-impedance circuit. For proper operation, it is recommended that V_{in} and V_{out} be constrained to the range $V_{SS} \le (V_{in}$ or $V_{out}) \le V_{DD}$. Reliability of operation is enhanced if unused inputs are connected to an appropriate logic voltage level (for instance, either V_{SS} or V_{DD}).

11.2 DC OPERATING CHARACTERISTICS

(V_{SS} = 0 V_{dc}, T_A = 25 °C)

Characteristic	Symbol	Min	Typ	Max	Unit
Operating Voltage external clock source f_{osc} = 2.0 MHz	V_{DD}	2.2	—	5.5	V

11.3 DC ELECTRICAL CHARACTERISTICS (5.0 Vdc)

(V_{DD} = 5.0 V_{dc} ±10%, V_{SS} = 0 V_{dc}, T_A = -40°C to +85 °C, unless otherwise noted)

Characteristic	Symbol	Min	Typ	Max	Unit
Output voltage I_{Load} = 10.0 µA I_{Load} = -10.0 µA	 V_{OL} V_{OH}	 — V_{DD} - 0.1	 — —	 0.1 —	 V
Output High Voltage (I_{Load} = -0.8 mA) PA0-7, PC0-7, PD0-7, PE0-7, PG0-7, PH0-7, PJ0-3	V_{OH}	V_{DD} - 0.8	—	—	V
Output Low Voltage (I_{Load} = 1.6 mA) PA0-7, PC0-7, PD0-7, PE0-7, PG0-7, PH0-7, PJ0-3	V_{OL}	—	—	0.4	V
Input High Voltage PA0-7, PB0-7, PC0-7, PD0-7, PE0-7, PF0-7, PG0-7, PH0-7, PJ0-3, OSC1, XOSC1, $\overline{RESET}$	V_{IH}	0.7 x V_{DD}	—	V_{DD}	V
Input Low Voltage PA0-7, PC0-7, PD0-7, PE0-7, PF0-7, PG0-7, PH0-7, PJ0-3, OSC1, XOSC1, $\overline{RESET}$	V_{IL}	V_{SS}	—	0.2 x V_{DD}	V
Input Low Voltage PB0-7	V_{IL}	V_{SS}	—	0.3 x V_{DD}	V
Supply Current (see Notes) Run Wait Stop (with XOSC operating) 25 °C -40 °C to +85 °C	 I_{DD} I_{DD} I_{DD} I_{DD}	 — — — —	 5.5 4.5 10 10	 10.0 6.0 20 20	 mA mA µA µA
I/O Ports Hi-Z Leakage Current PA0-7, PC0-7, PD0-7, PE0-7, PG0-7, PH0-7	I_{OZ}	—	—	10	µA
Input Current $\overline{RESET}$, OSC1	I_{in}	—	—	1	µA
Capacitance Ports (as Input or Output) $\overline{RESET}$	 C_{out} C_{in}	 — —	 — —	 12 8	 pF pF

NOTES:
6. These I_{DD} values are design goals and do not reflect characterization data.

11.4 DC ELECTRICAL CHARACTERISTICS (2.5 Vdc)

(V_{DD} = 2.5 V_{dc} ±10%, V_{SS} = 0 V_{dc}, T_A = -40 °C to +85 °C, unless otherwise noted)

Characteristic	Symbol	Min	Typ	Max	Unit
Output Voltage I_{Load} = 10.0 µA I_{Load} = -10.0 µA	 V_{OL} V_{OH}	 — V_{DD} - 0.1	 — —	 0.1 —	 V
Output High Voltage (I_{Load} = -0.8 mA) PA0-7, PC0-7, PD0-7, PE0-7, PG0-7, PH0-7, PJ0-3	V_{OH}	V_{DD} - 0.8	—	—	V
Output Low Voltage (I_{Load} = 1.6 mA) PA0-7, PC0-7, PD0-7, PE0-7, PG0-7, PH0-7, PJ0-3	V_{OL}	—	—	0.3	V
Input High Voltage PA0-7, PB0-7, PC0-7, PD0-7, PE0-7, PF0-7, PG0-7, PH0-7, PJ0-3, OSC1, XOSC1, $\overline{RESET}$	V_{IH}	0.7 x V_{DD}	—	V_{DD}	V
Input Low Voltage PA0-7, PC0-7, PD0-7, PE0-7, PF0-7, PG0-7, PH0-7, PJ0-3, OSC1, XOSC1, $\overline{RESET}$	V_{IL}	V_{SS}	—	0.2 x V_{DD}	V
Input Low Voltage PB0-7	V_{IL}	V_{SS}	—	0.3 x V_{DD}	V
Supply Current (See Notes) Run Wait Stop (With XOSC Operating) 25 °C -40 °C to +85 °C	 I_{DD} I_{DD} I_{DD} I_{DD}	 — — — —	 1.5 3.5 5 5	 5.0 4.0 10 10	 mA mA µA µA
I/O Ports Hi-Z Leakage Current PA0-7, PC0-7, PD0-7, PE0-7, PG0-7, PH0-7	I_{OZ}	—	—	10	µA
Input Current $\overline{RESET}$, OSC1	I_{in}	—	—	1	µA
Capacitance Ports (As Input or Output) $\overline{RESET}$	 C_{out} C_{in}	 — —	 — —	 12 8	 pF pF

NOTES:
1. These I_{DD} values are design goals and do not reflect characterization data.

11.5 A/D CONVERTER CHARACTERISTICS

(V_{DD} = 5.0 V_{dc} ±10%, V_{SS} = 0 V_{dc}, T_A = -40 °C to +85 °C, unless otherwise noted)

Characteristic	Min	Max	Unit	Comments
Resolution	8	8	Bits	
Absolute Accuracy (V_{REFL} = 0 V, V_{REFH} = 4.0-V_{DD})	—	±1 1/2	LSB	Including quantization
Conversion Range V_{REFH} V_{REFL}	V_{REFL} V_{REFL} -0.1	V_{REFH} V_{DD} V_{REFH}	V V V	A/D accuracy may decrease proportionately as V_{REFH} is reduced below 4.0
Power-up Time	—	100	µs	
Input Leakage PF0-PF7 V_{REFL},V_{REFH}	 — —	 ±400 ±1	 nA µA	
Conversion Time (Includes Sampling Time)	32	32	T_{AD}*	
Monotonicity	Inherent (Within Total Error)			
Zero Input Reading	00	01	Hex	V_{in} = 0V
Ratiometric Reading	FF	FF	Hex	V_{in} = V_{REFH}
Sample Time	12	12	T_{AD}*	
Input Capacitance	—	8	pF	
Analog Input Voltage	V_{REFL}	V_{REFH}	V	

*T_{AD} = t_{cyc} if clock source equals MCU.

11.6 CONTROL TIMING (5.0 Vdc)

(V_{DD} = 5.0 Vdc ±10%, V_{SS} = 0 Vdc, T_A = -40 °C to +85 °C, unless otherwise noted)

Characteristic	Symbol	Min	Max	Unit
Frequency of Operation Crystal Option External Clock Option	 f_{osc} f_{osc}	 — dc	 4.0 4.0	 MHz MHz
Internal Operating Frequency Crystal (f_{osc} ÷ 2) External Clock (f_{osc} ÷ 2)	 f_{op} f_{op}	 — dc	 2.0 2.0	 MHz MHz
Cycle Time	t_{cyc}	500	—	ns
Crystal Oscillator Startup Time	t_{OXOV}	—	100	ms
Stop Recovery Startup Time (Crystal Oscillator)	t_{ILCH}	—	100	ms
$\overline{RESET}$ Pulse Width	t_{RL}	1.5	—	t_{cyc}
Interrupt Pulse Width Low (Edge-Triggered)	t_{ILIH}	125	—	ns
Interrupt Pulse Period	t_{ILIL}	†	—	t_{cyc}
OSC1 Pulse Width	t_{OH},t_{OL}	100	—	ns
A/D On Current Stabilization Time	t_{ADON}	—	100	µs
RC Oscillator Stabilization Time (A/D)	t_{RCON}	—	5.0	µs

† The minimum period t_{ILIL} should not be less than the number of cycle times it takes to execute the interrupt service routine plus 21 t_{cyc}.

11.7 CONTROL TIMING (2.5 Vdc)

(V_{DD} = 2.5 V_{dc} ±10%, V_{SS} = 0 V_{dc}, T_A = -40 °C to +85 °C, unless otherwise noted)

Characteristic	Symbol	Min	Max	Unit
Frequency of Operation Crystal Option External Clock Option	 f_{osc} f_{osc}	 — dc	 2.0 2.0	 MHz MHz
Internal Operating Frequency Crystal (f_{osc} ÷ 2) External Clock (f_{osc} ÷ 2)	 f_{op} f_{op}	 — dc	 1.0 1.0	 MHz MHz
Cycle Time	t_{cyc}	1000	—	ns
Crystal Oscillator Start-up Time	t_{OXOV}	—	200	ms
Stop Recovery Start-up Time (Crystal Oscillator)	t_{ILCH}	—	200	ms
$\overline{\text{RESET}}$ Pulse Width	t_{RL}	1.5	—	t_{cyc}
Interrupt Pulse Width Low (Edge-Triggered)	t_{ILIH}	250	—	ns
Interrupt Pulse Period	t_{ILIL}	†	—	t_{cyc}
OSC1 Pulse Width	t_{OH},t_{OL}	200	—	ns

† The minimum period t_{ILIL} should not be less than the number of cycle times it takes to execute the interrupt service routine plus 21 t_{cyc}.

Home Page:
www.freescale.com
email:
support@freescale.com
USA/Europe or Locations Not Listed:
Freescale Semiconductor
Technical Information Center, CH370
1300 N. Alma School Road
Chandler, Arizona 85224
(800) 521-6274
480-768-2130
support@freescale.com
Europe, Middle East, and Africa:
Freescale Halbleiter Deutschland GmbH
Technical Information Center
Schatzbogen 7
81829 Muenchen, Germany
+44 1296 380 456 (English)
+46 8 52200080 (English)
+49 89 92103 559 (German)
+33 1 69 35 48 48 (French)
support@freescale.com
Japan:
Freescale Semiconductor Japan Ltd.
Headquarters
ARCO Tower 15F
1-8-1, Shimo-Meguro, Meguro-ku
Tokyo 153-0064, Japan
0120 191014
+81 2666 8080
support.japan@freescale.com
Asia/Pacific:
Freescale Semiconductor Hong Kong Ltd.
Technical Information Center
2 Dai King Street
Tai Po Industrial Estate,
Tai Po, N.T., Hong Kong
+800 2666 8080
support.asia@freescale.com
For Literature Requests Only:
Freescale Semiconductor
Literature Distribution Center
P.O. Box 5405
Denver, Colorado 80217
(800) 441-2447
303-675-2140
Fax: 303-675-2150
LDCForFreescaleSemiconductor
@hibbertgroup.com

RoHS-compliant and/or Pb- free versions of Freescale products have the functionality and electrical characteristics of their non-RoHS-compliant and/or non-Pb- free counterparts. For further information, see http://www.freescale.com or contact your Freescale sales representative.

For information on Freescale.s Environmental Products program, go to http://www.freescale.com/epp.