MC68HC05F8

HC05

MC68HC05F8
MC68HC705F8

TECHNICAL
DATA

TECHNICAL DATA

MOTOROLA

MOTOROLA

MC68HC05F8
MC68HC705F8

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

Conventions

Register and bit mnemonics are defined in the paragraphs describing them.

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

Unless otherwise stated, blank cells in a register diagram indicate that the bit is either unused or reserved; shaded cells indicate that the bit is not described in the following paragraphs; ‘u’ is used to indicate an undefined state (on reset).

CUSTOMER FEEDBACK QUESTIONNAIRE (MC68HC05F8D/H)

Motorola wishes to continue to improve the quality of its documentation. We would welcome your feedback on the publication you have just received. Having used the document, please complete this card (or a photocopy of it, if you prefer).

1. How would you rate the quality of the document? Check one box in each category.

	Excellent			Poor		Excellent			Poor
Organization	☐	☐	☐	☐	Tables	☐	☐	☐	☐
Readability	☐	☐	☐	☐	Table of contents	☐	☐	☐	☐
Understandability	☐	☐	☐	☐	Index	☐	☐	☐	☐
Accuracy	☐	☐	☐	☐	Page size/binding	☐	☐	☐	☐
Illustrations	☐	☐	☐	☐	Overall impression	☐	☐	☐	☐

Comments: _____

2. What is your intended use for this document? If more than one option applies, please rank them (1, 2, 3).

Selection of device for new application ☐
System design ☐
Training purposes ☐
Other ☐ Please specify: _____

3. How well does this manual enable you to perform the task(s) outlined in question 2?

Completely ☐ ☐ ☐ ☐ Not at all

Comments: _____

4. How easy is it to find the information you are looking for?

Easy ☐ ☐ ☐ ☐ Difficult

Comments: _____

5. Is the level of technical detail in the following sections sufficient to allow you to understand how the device functions?

		Too little detail				Too much detail
SECTION 1	GENERAL DESCRIPTION	☐	☐	☐	☐	☐
SECTION 2	PIN DESCRIPTIONS	☐	☐	☐	☐	☐
SECTION 3	MEMORY AND REGISTERS	☐	☐	☐	☐	☐
SECTION 4	RESETS	☐	☐	☐	☐	☐
SECTION 5	INTERRUPTS	☐	☐	☐	☐	☐
SECTION 6	TIMERS	☐	☐	☐	☐	☐
SECTION 7	SERIAL PERIPHERAL INTERFACE	☐	☐	☐	☐	☐
SECTION 8	MANCHESTER ENCODER/DECODER	☐	☐	☐	☐	☐
SECTION 9	DTMF/MELODY GENERATOR	☐	☐	☐	☐	☐
SECTION 10	CPU CORE AND INSTRUCTION SET	☐	☐	☐	☐	☐
SECTION 11	LOW POWER MODES	☐	☐	☐	☐	☐
SECTION 12	OPERATING MODES	☐	☐	☐	☐	☐
SECTION 13	ELECTRICAL SPECIFICATIONS	☐	☐	☐	☐	☐
SECTION 14	MECHANICAL SPECIFICATIONS	☐	☐	☐	☐	☐

Comments: _____

6. Have you found any errors? If so, please comment: _____

7. From your point of view, is anything missing from the document? If so, please say what: _____

– Cut along this line to remove –

8. How could we improve this document? _____

_____

_____

9. How would you rate Motorola's documentation?

	Excellent			Poor
– In general	❑	❑	❑	❑
– Against other semiconductor suppliers	❑	❑	❑	❑

10. Which semiconductor manufacturer provides the best technical documentation? _____

11. Which company (in any field) provides the best technical documentation? _____

12. How many years have you worked with microprocessors?

Less than 1 year ❑ 1–3 years ❑ 3–5 years ❑ More than 5 years ❑

– Second fold back along this line –

By air mail
Par avion

FIX STAMP HERE

– First fold back along this line –

– Cut along this line to remove –

Motorola Semiconductors H.K. Ltd.,
13/F, Prosperity Centre,
77-81 Container Port Road,
Kwai Chung, N.T.,
HONG KONG.

F.A.O. HKG CSIC Technical Publications
(re: MC68HC05F8D/H)

FAX: (852) 2485-0548

MOTOROLA
Semiconductor Products Sector
Asia Pacific Group

– Third fold back along this line –

13. Currently there is some discussion in the semiconductor industry regarding a move towards providing data sheets in electronic form. If you have any opinion on this subject, please comment. _____

_____

14. We would be grateful if you would supply the following information (at your discretion), or attach your card.

Name: _____ Phone No: _____

Position: _____ FAX No: _____

Department: _____

Company: _____

Address: _____

_____

Thank you for helping us improve our documentation,
HKG CSIC Technical Publications , Motorola Semiconductors H.K. Ltd., Hong Kong

– Finally, tuck this edge into opposite flap –

TABLE OF CONTENTS

5
INTERRUPTS

6
TIMERS

7
SERIAL PERIPHERAL INTERFACE

8
MANCHESTER ENCODER/DECODER

9
DTMF/MELODY GENERATOR

10
CPU CORE AND INSTRUCTION SET

11
LOW POWER MODES

12
OPERATING MODES

13
ELECTRICAL SPECIFICATIONS

14
MECHANICAL SPECIFICATIONS

THIS PAGE LEFT BLANK INTENTIONALLY

LIST OF FIGURES

THIS PAGE LEFT BLANK INTENTIONALLY

LIST OF TABLES

THIS PAGE LEFT BLANK INTENTIONALLY

1
GENERAL DESCRIPTION

The MC68HC05F8 HCMOS microcontroller is a member of the M68HC05 family of low-cost single-chip microcontrollers. This 8-bit microcontroller unit (MCU) contains an on-chip oscillator, CPU, RAM, ROM, I/O, timer, serial peripheral interface, Manchester encoder/decoder, DTMF/melody generator, and COP watchdog monitor. This MCU is particularly suitable for cordless telephones with an answering machine.

The MC68HC705F8 is an EPROM version of the MC68HC05F8. All references to the MC68HC05F8 apply equally to the MC68HC705F8, unless otherwise stated. *References specific to the MC68HC705F8 are italicized in the text.*

1.1 Features

The following are some of the hardware and software features of the MC68HC05F8 single-chip microcontroller.

1.1.1 Hardware Features

- HCMOS technology
- 8-bit architecture
- Power saving Wait and Stop modes
- Full static operation
- 2.5V to 6V operating voltage
- 320 bytes of on-chip RAM (64 bytes for stack)
- 8K-bytes of on-chip ROM; *8K-bytes of on-chip EPROM for MC68HC705F8*
- 496-bytes self-check ROM; *496-bytes bootstrap ROM for MC68HC705F8*
- 8 keyboard interrupt lines

- Manchester encoder/decoder
- DTMF/melody generator
- Oscillator for 3.579 MHz crystal
- 16-bit free-running programmable timer with 4 selectable prescaler frequencies
- 16-bit auto-reload timer with 4 selectable prescaler frequencies
- Computer Operating Properly (COP) watchdog monitor
- Serial peripheral interface
- 10mA high current output pins for LED direct driving
- Available in 56-pin SDIP and 64-pin QFP packages

1.1.2 Software Features

- Similar to MC6800
- 8 x 8 unsigned multiply instruction
- Efficient use of program space
- Versatile interrupt handling
- True bit manipulation
- Addressing modes with index addressing for tables
- Efficient instruction set
- Memory mapped I/O
- Two power saving standby modes
- Upward software compatible with the M146805 CMOS family

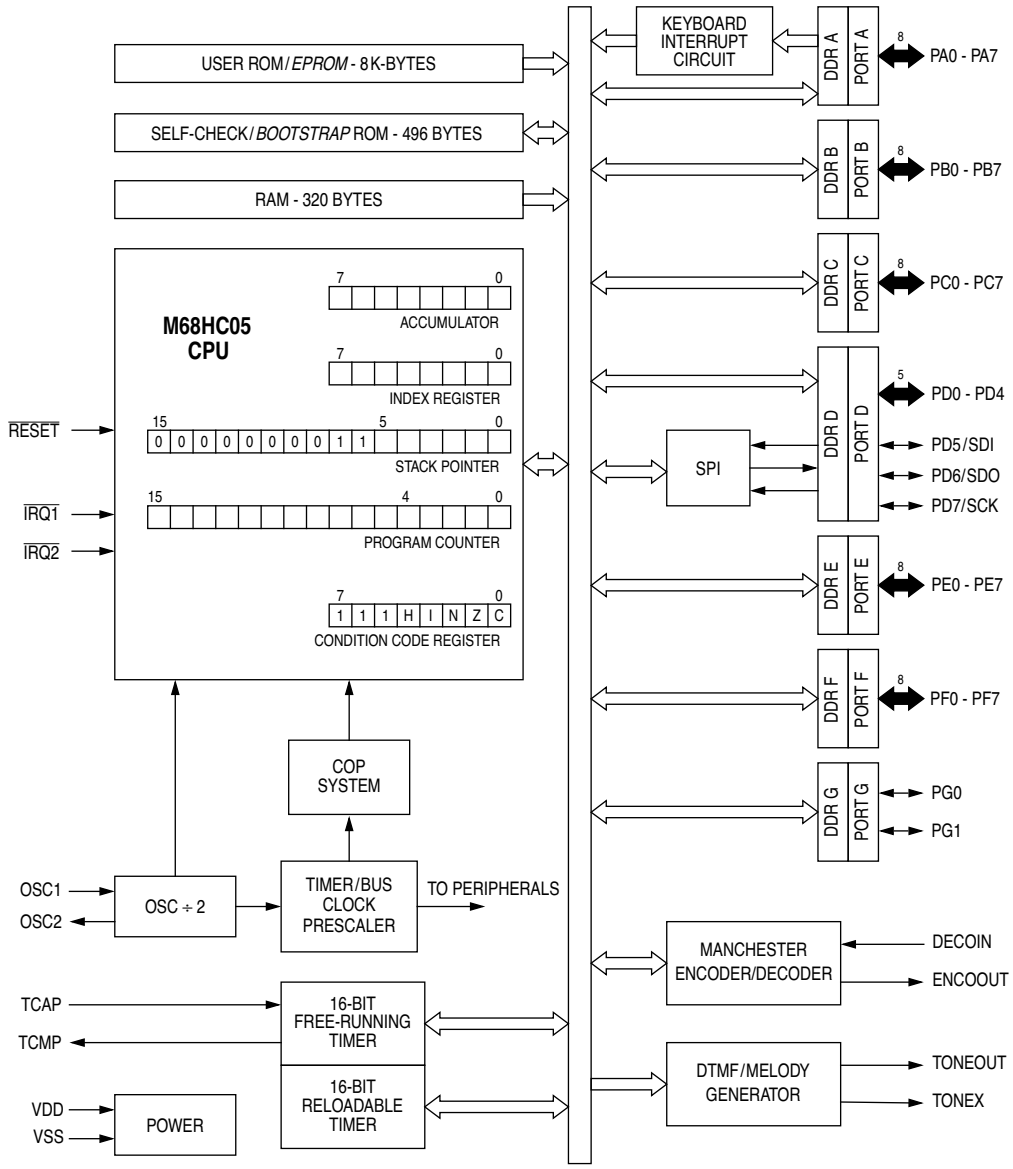


Figure 1-1 MC68HC05F8/MC68HC705F8 Block Diagram

THIS PAGE LEFT BLANK INTENTIONALLY

2
PIN DESCRIPTIONS

This section provides a description of the functional pins and I/O programming of the MC68HC05F8/*MC68HC705F8* microcontroller.

2.1 Functional Pin Descriptions

PIN NAME	56-pin SDIP PIN No.	64-pin QFP PIN No.	DESCRIPTION
VDD, VSS	53, 52	52, 51	Power is supplied to the MCU using these two pins. VDD is power and VSS is ground.
$\overline{IRQ1}$ $\overline{IRQ2}$	10 9	2 1	$\overline{IRQ1}$ and $\overline{IRQ2}$ are software programmable to provide two choices of interrupt triggering sensitivity. These options are: 1) negative edge-sensitive triggering only, or 2) both negative edge-sensitive and level sensitive triggering.
$\overline{RESET}$	34	30	The active low $\overline{RESET}$ input is not required for start-up, but can be used to reset the MCU internal state and provide an orderly software start-up procedure.
TCAP	49	48	The TCAP input controls the input capture feature for the on-chip programmable free-running timer.
TCMP	48	47	The TCMP pin provides an output for the output compare feature of the on-chip programmable free-running timer.
OSC1, OSC2	33, 32	29, 28	These pins provide connections to the on-chip oscillator. The crystal frequency is 3.579545 MHz. OSC1 may be driven by an external oscillator if an external crystal circuit is not used.
PA0-PA7	(PA4-PA7 only) 14-11	10-3	These eight I/O lines comprise port A. The state of any pin is software programmable. All port A lines are configured as input during power on or external reset. Port A can also be programmed as keyboard interrupts. PA0 to PA3 are not bonded out on the 56-pin package.
PB0-PB7	22-15	18-11	These eight I/O lines comprise port B. The state of any pin is software programmable. All port B lines are configured as input during power on or external reset.
PC0-PC7	5-1, 56-54	60-53	These eight I/O lines comprise port C. The state of any pin is software programmable. All port C lines are configured as input during power on or external reset. Each port C pins also has the ability to sink a maximum current of 10 mA with a maximum saturation of 1 V.

PIN NAME	56-pin SDIP PIN No.	64-pin QFP PIN No.	DESCRIPTION
PD0-PD7 SDI SDO SCK	(PD0-PD4 only) 47-43 - - -	46-39 41 40 39	These eight I/O lines comprise port D. The state of any pin is software programmable. All port D lines are configured as input during power on or external reset. When the SPE bit of the SPI Control register (bit 6 of address $10) is set, PD5, PD6, & PD7 are used for SDI, SDO, & SCK respectively, for the Serial Peripheral Interface. PD5 to PD7 are not bonded out on the 56-pin package. Hence, the 56-pin package does not have SPI features.
PE0-PE7	30-23	26-19	These eight I/O lines comprise port E. The state of any pin is software programmable. All port E lines are configured as input during power on or external reset.
PF0-PF7	42-35	38-31	These eight I/O lines comprise port F. The state of any pin is software programmable. All port F lines are configured as input during power on or external reset.
PG0, PG1	(PG0 only) 8	64, 63	These two I/O lines comprise port G. The state of any pin is software programmable. All port G lines are configured as input during power on or external reset. PG1 is not bonded out on the 56-pin package.
ENCOOUT	6	61	This pin is for encoded data output from the Manchester encoder.
DECOIN	7	62	This pin is for raw (Manchester) data input to the Manchester decoder.
TONEOUT	50	49	This output pin provides dual tone DTMF or melody under the control of the DTMF/Melody Generator.
TONEX	51	50	This output pin provides pacifier tones under the control of the DTMF/Melody Generator.
NC/*VPP*	31	27	This pin is used as the programming voltage pin for the EPROM version, MC68HC705F8. It is connected to VDD for normal operation. This pin is not used in the standard ROM part, MC68HC05F8.

2.2 Pin Assignments

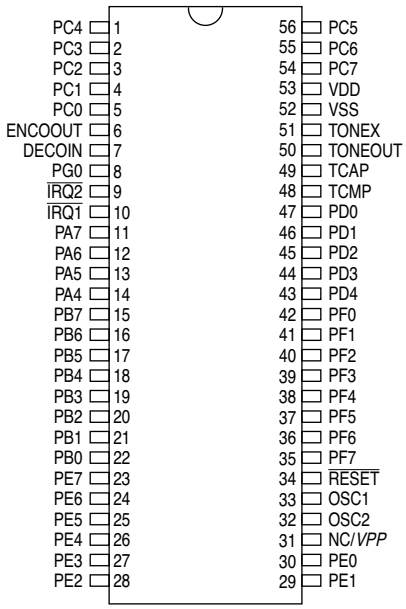


Figure 2-1 Pin Assignments for 56-pin SDIP package

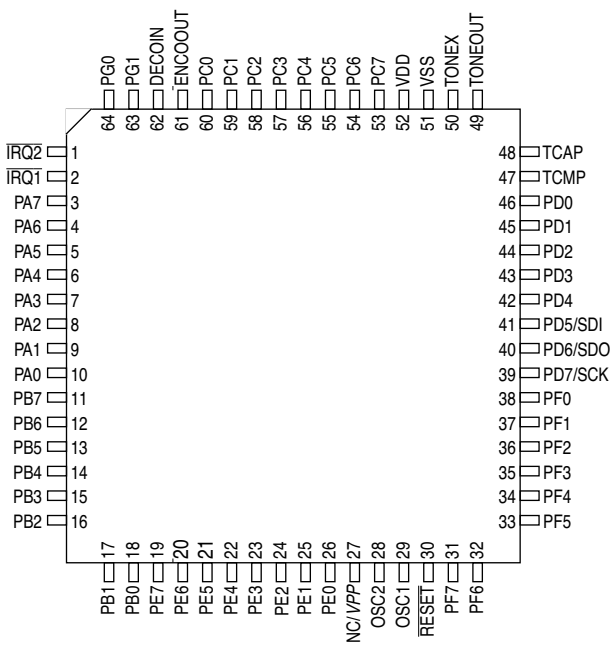


Figure 2-2 Pin Assignments for 64-pin QFP package

2.3 Input/Output Programming

2.3.1 Parallel Ports

Port A, B, C, D, E, F and G may be programmed as an input or an output under software control. The direction of the pins is determined by the state of corresponding bit in the port data direction register (DDR). Each 8-bit port (except port G, where it has only 2 bits) has an associated 8-bit data direction register. Any port A, B, C, D, E, F or G pin is configured as an output if its corresponding DDR bit is set to a logic one. A pin is configured as an input if its corresponding DDR bit is cleared to a logic zero. At power-on or reset, all DDRs are cleared, which configure all port A, B, C, D, E, F and G pins as inputs. The data direction registers are capable of being written to or read by the processor. Refer to Figure 2-3 and Table 2-1. During the programmed output state, a read of the data register actually reads the value of the output data latch and not the I/O pin.

Table 2-1 I/O Pin Functions

R/$\overline{W}$	DDR	I/O Pin Function
0	0	The I/O pin is in input mode. Data is written into the output data latch.
0	1	Data is written into the output data latch and output to the I/O pin.
1	0	The state of the I/O pin is read.
1	1	The I/O pin is in an output mode. The output data latch is read.

2.3.2 Serial Port (SPI)

The serial peripheral interface (SPI) uses the port D pins for its function. The SPI function requires three of the pins (PD5-PD7) for its serial data input (SDI), serial data output (SDO), and system clock (SCK) respectively. See Section 7 for detailed description of SPI.

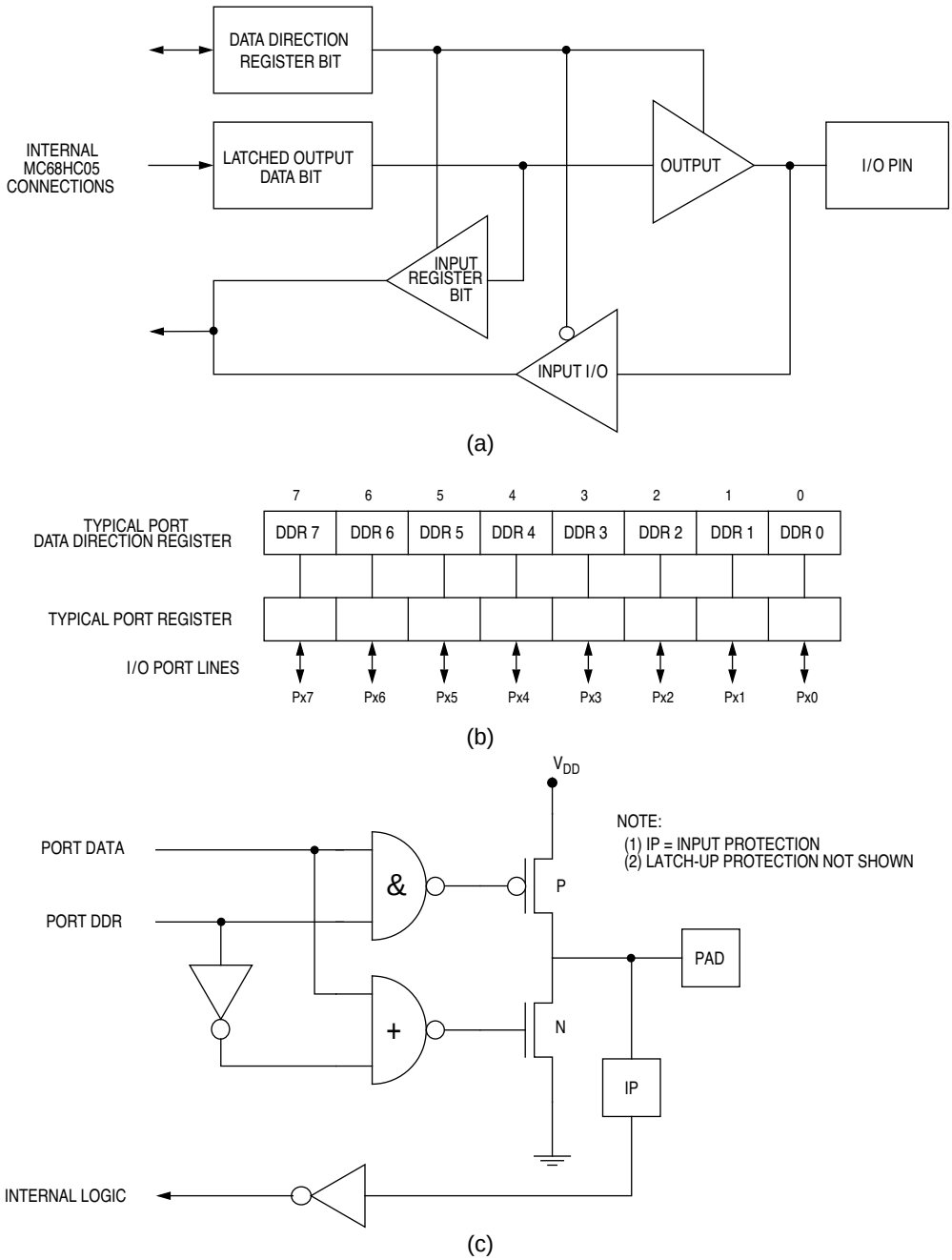


Figure 2-3 Parallel Port I/O Circuitry

THIS PAGE LEFT BLANK INTENTIONALLY

3
MEMORY AND REGISTERS

This section describes the organization of the on-chip memory.

3.1 Memory Map

The CPU can address 64K-bytes of memory space. The ROM portion of memory holds the program instructions, fixed data, user-defined vectors, and interrupt service routines. The RAM portion of memory holds variable data. I/O registers are memory-mapped so that the CPU can access their locations in the same way that it accesses all other memory locations. Figure 3-1 shows the Memory Map for the MC68HC05F8/*MC68HC705F8*.

3.2 Input/Output Section

The first 64 addresses of memory space, $0000-$003F, are the I/O section. These are the addresses of the I/O control registers, status registers, and data registers.

3.3 RAM

The 320 addresses from $0040-$017F are RAM locations. The CPU uses the 64 RAM addresses, $00C0-$00FF, as the stack. Before processing an interrupt, the CPU uses five bytes of the stack to save the contents of the CPU registers. During a subroutine call, the CPU uses two bytes of the stack to store the return address. The stack pointer decrements during pushes and increments during pulls.

Note: Be careful when using nested subroutines or multiple interrupt levels. The CPU may overwrite data in the RAM during a subroutine or during the interrupt stacking operation. Once the stack pointer passes $00C0, it wraps round back to $00FF.

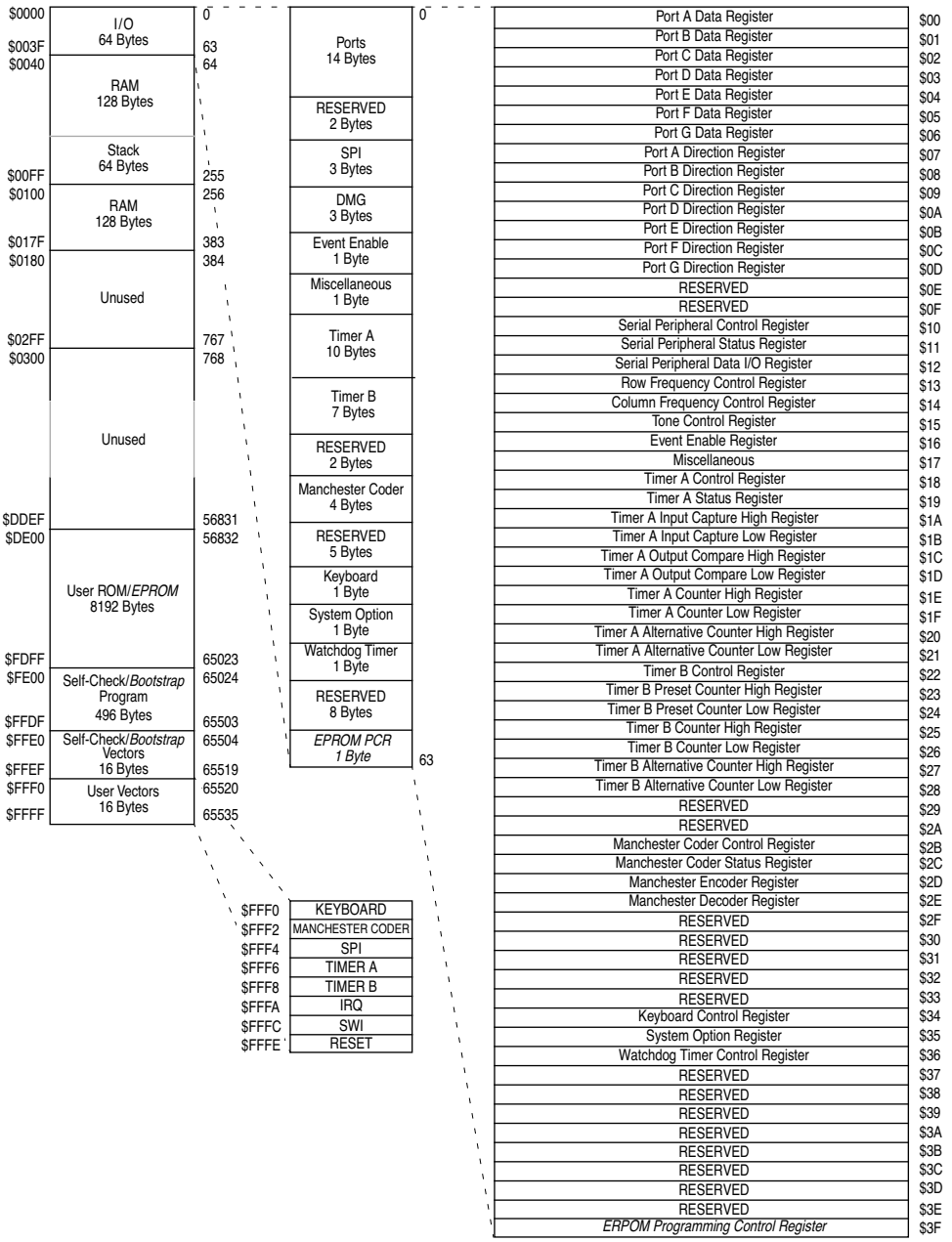


Figure 3-1 MC68HC05F8/*MC68HC705F8* Memory Map

Table 3-1 MC68HC05F8/*MC68HC705F8* Registers

Address	Register Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
$00	Port A data	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
$01	Part B data	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
$02	Port C data	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
$03	Port D data	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
$04	Port E data	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
$05	Port F data	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
$06	Port G data							bit 1	bit 0
$07	Port A data direction	DDR7	DDR6	DDR5	DDR4	DDR3	DDR2	DDR1	DDR0
$08	Port B data direction	DDR7	DDR6	DDR5	DDR4	DDR3	DDR2	DDR1	DDR0
$09	Port C data direction	DDR7	DDR6	DDR5	DDR4	DDR3	DDR2	DDR1	DDR0
$0A	Port D data direction	DDR7	DDR6	DDR5	DDR4	DDR3	DDR2	DDR1	DDR0
$0B	Port E data direction	DDR7	DDR6	DDR5	DDR4	DDR3	DDR2	DDR1	DDR0
$0C	Port F data direction	DDR7	DDR6	DDR5	DDR4	DDR3	DDR2	DDR1	DDR0
$0D	Port G data direction							DDR1	DDR0
$0E	Not used								
$0F	Not used								
$10	SPI control	SPIE	SPE		MSTR				
$11	SPI status	SPIF	DCOL						
$12	SPI data I/O								
$13	Row frequency control				FCR4	FCR3	FCR2	FCR1	FCR0
$14	Column frequency control				FCC4	FCC3	FCC2	FCC1	FCC0
$15	Tone control	MS1	MS0	TGER	TGEC				
$16	Event enable	TIMH	INTE1	INTE2					
$17	Miscellaneous	POR	INTF1	INTF2	KEYF				
$18	Timer A control	ICIE	OCIE	TOIE				IEDG	OLVL
$19	Timer A status	ICF	OCF	TOF					
$1A	Timer A input capture high								
$1B	Timer A input capture low								
$1C	Timer A output compare high								
$1D	Timer A output compare low								
$1E	Timer A counter high								
$1F	Timer A counter low								

Table 3-1 MC68HC05F8/*MC68HC705F8* Registers

Address	Register Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
$20	Timer A alternative counter high								
$21	Timer A alternative counter low								
$22	Timer B control	TMBE	TBOIE				TCSB1	TCSB0	TUF
$23	Timer B preset counter high								
$24	Timer B preset counter low								
$25	Timer B counter high								
$26	Timer B counter low								
$27	Timer B alternative counter high								
$28	Timer B alternative counter low								
$29	Not used								
$2A	Not used								
$2B	Manchester coder control	NCE	NIE	CIE	DCE	DIE		BR1	BR0
$2C	Manchester coder status	NCM	NCC	DCF	OVF				
$2D	Manchester encoder data								
$2E	Manchester decoder data								
$2F	Not used								
$30	Reserved								
$31	Not used								
$32	Not used								
$33	Not used								
$34	Keyboard control	KEYE				KEYX7	KEYX6	KEYX5	KEYX4
$35	System option		TCSA1	TCSA0	INTN1	INTN2			
$36	Watchdog timer control	WDTE	WDTE	KWDT	WDTOF			WDT1	WDT0
$37	Not used								
$38	Not used								
$39	Not used								
$3A	Not used								
$3B	Not used								
$3C	Reserved								
$3D	Reserved								
$3E	Not used								
$3F	*EPROM programming control*							LAT	EPGM

4
RESETS

The MC68HC05F8 can be reset in four ways: by the initial power-on reset function, by an active low input to the $\overline{\text{RESET}}$ pin, by an opcode fetch from an illegal address, and by a COP watchdog reset (if the watchdog timer is enabled). Any of these resets will cause the program to go to its starting address, specified by the contents of memory locations $FFFE and $FFFF, and cause the interrupt mask of the Condition Code register to be set.

4.1 Power-On Reset (POR)

The power-on reset occurs when a positive transition is detected on the supply voltage, V_{DD}. The power-on reset is used strictly for power-up conditions, and should not be used to detect any drops in the power supply voltage. There is no provision for a power-down reset. The power-on circuitry provides for a 4064 tcyc delay from the time that the oscillator becomes active. If the external $\overline{\text{RESET}}$ pin is low at the end of the 4064 tcyc time out, the processor remains in the reset condition until $\overline{\text{RESET}}$ goes high. The user must ensure that V_{DD} has risen to a point where the MCU can operate properly prior to the time the 4064 POR cycles have elapsed. If there is doubt, the external $\overline{\text{RESET}}$ pin should remain low until such time that V_{DD} has risen to the minimum operating voltage specified.

After a power-on reset the POR bit in the Miscellaneous register (bit 7 of address $17) is set, indicating the reset was cause by a power-on, not COP watchdog time-out or external reset. The POR bit is cleared by writing a logic “0” to the bit. The POR cannot be set by software.

4.2 $\overline{\text{RESET}}$ Pin

The $\overline{\text{RESET}}$ input pin is used to reset the MCU to provide an orderly software start-up procedure. When using the external reset, the $\overline{\text{RESET}}$ pin must stay low for a minimum of 1.5tcyc. The $\overline{\text{RESET}}$ pin contains an internal Schmitt Trigger as part of its input to improve noise immunity.

4.3 Illegal Address (ILADR) Reset

The MCU monitors all opcode fetches. If an illegal address space is accessed during an opcode fetch, an internal reset is generated. Illegal address spaces consist of all unused locations within the memory map and the I/O registers (see Figure 3-1). Because the internal reset signal is used, the MCU comes out of an ILADR reset in the same operating mode it was in when the opcode was fetched.

4.4 Computer Operating Properly (COP) Reset

The MCU contains a watchdog timer that automatically times out if not reset (cleared) within a specific amount of time by a program reset sequence.

Note: COP time-out is prevented by periodically writing a logic 1 to bit 7 of address $36.

If the watchdog timer is allowed to time-out, an internal reset is generated to reset the MCU. Because the internal reset signal is used, the MCU comes out of a COP reset in the same operating mode it was in when the COP time-out was generated.

The watchdog timer is initially disabled after a reset, it is enabled by writing a '1' to bit 7 of address $36. Once enabled, it cannot be disabled by software.

Refer to Section 6.3 for detailed description of the COP watchdog system.

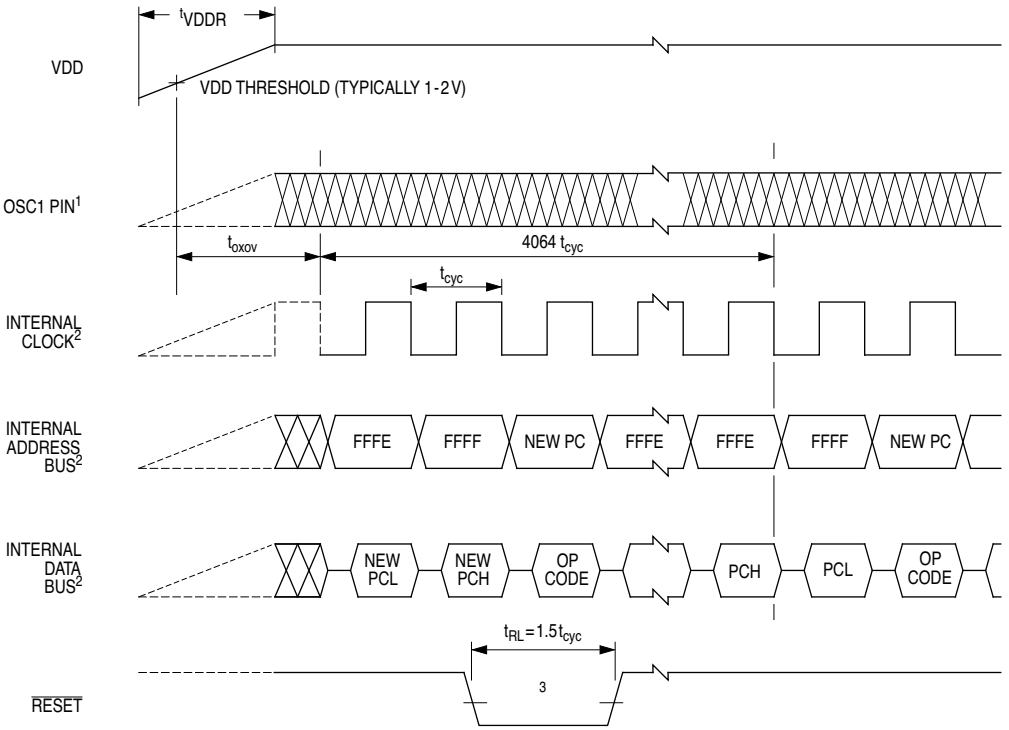
Table 4-1 shows the internal circuit actions on reset, but not necessary in order of occurrence.

Table 4-1 Reset Action on Internal Circuit

	DEFAULT CONDITIONS AFTER RESET
1	Timer A not inhibited (TIMHA bit cleared).
2	Timer A prescaler reset to zero state.
3	Timer A counter configures to $FFFC.
4	Timer A output compare (TCMP) bit is reset to zero.
5	Timer A clock=internal bus clock ÷ 4 (TCSA1=TCSA0=0).
6	All timer A interrupt enable bits cleared (ICIE, OCIE, and TOIE) to disable timer interrupts. The OLVL timer bit is also cleared by reset.
7	Timer B is disabled (TMBE bit cleared).
8	Timer B prescaler reset to zero (TCSB0=TCSB1=0).
9	All data direction registers cleared to zero (default as inputs).
10	Stack pointer configured to $00FF.
11	Internal address bus forced to restart vector ($FFFE-$FFFF).
12	I bit of condition code register set to logic 1.
13*	STOP latch cleared.
14	WAIT latch cleared.
15	External interrupt latch cleared (INTF1=INTF2=0).
16	External interrupt enable bits cleared (INTE1 & INTE2).
17	SPI disabled (serial output enable control bit SPE=0). Other SPI bits cleared are SPIE, MSTR, SPIF, and DCOL.
18	SPI system configured to slave mode (MSTR=0).
19	Keyboard interrupt enabled (KEYE) and keyboard interrupt flag (KEYF) bits are cleared.
20	Disable MANCD (NCD=DCE=0).
21	Disable tone generation in DMG (TGER=TGEC=0).
22	Place DMG in DTMF mode (MS1=MS0=0).
23	Watchdog timer is inhibited (WDTE=0), kill function is disabled (KWDT=0).
24	If reset is by POR, set POR bit.

* Indicates that time-out still occurs.

Listed numbers do not represent order of occurrence.



NOTES:
1. OSC1 is not meant to represent frequency. It is only used to represent time.
2. Internal clock, internal address bus, and internal data bus signals are not available externally.
3. Next rising edge of internal clock after rising edge of $\overline{\text{RESET}}$ initiates reset sequence.

Figure 4-1 Power-On Reset and $\overline{\text{RESET}}$ Timing

5
INTERRUPTS

The MC68HC05F8 is capable of handling eight types of interrupt, seven hardware and one software. The interrupt mask bit (“I” bit in the Condition Code register), if set, masks all interrupts except the software interrupt, SWI. Interrupts such as IRQ, Timers, and MANCD have several flags which will cause the interrupt. Interrupt flags are found in “read only” status registers, while their enables are in associated control registers. They are never mixed in the same register. If the enable bit is “0”, it masks the interrupt from occurring but does not inhibit the flag from being set. A reset clears all enable bits. The general sequence for clearing an interrupt is a software sequence of reading the status register while the flag is set followed by a read or write of an associated register. When any of these interrupts occur, and if enabled, normal processing is suspended at the end of the current instruction execution. The state of the machine is pushed onto the stack (see Figure 5-1 for stacking order) and the appropriate vector points to the starting address of the interrupt service routine (see Table 5-1). Also, the interrupt mask bit in the condition code register is set. This masks further interrupts. At the completion of the service routine, the software normally contains an RTI instruction which, when executed, restores the machine state and continues executing the interrupted program. Figure 5-2 is a flowchart showing the program flow and interrupt priority for hardware interrupts.

Note: The interrupt mask bit (I bit) will be cleared if and only if the corresponding bit stored on the stack is zero.

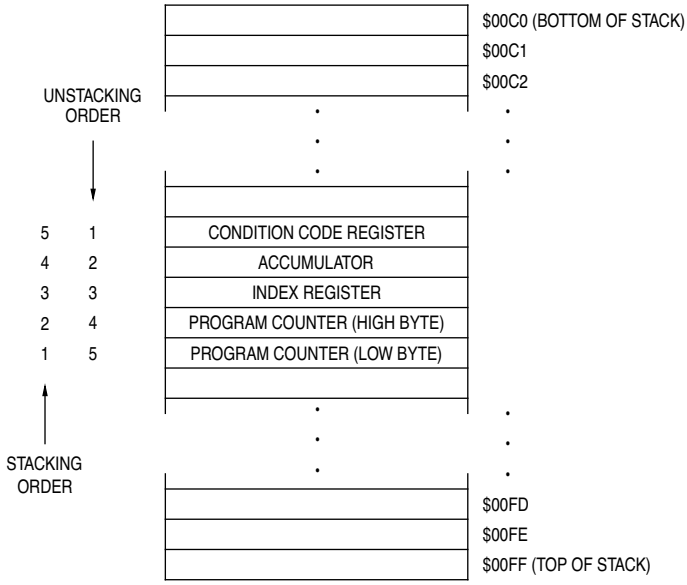


Figure 5-1 Interrupt Stacking Order

Table 5-1 Reset/Interrupt Vector Addresses

Register	Flag Name	Interrupt	CPU Interrupt	Vector Address
–	–	Reset	$\overline{\text{RESET}}$	$FFFE-$FFFF
–	–	Software	SWI	$FFFC-$FFFD
Miscellaneous	INTF 1	External Interrupt 1	$\overline{\text{IRQ}}$	$FFFA-$FFFB
	INTF 2	External Interrupt 2		
Timer B Control & Status	TUF	Timer Underflow	TIMER B	$FFF8-$FFF9
Timer A Status	ICF	Input Capture	TIMER A	$FFF6-$FFF7
	OCF	Output Compare		
	TOF	Timer Overflow		
SPI Status	SPIF	SPI Interrupt	SPI	$FFF4-$FFF5
MANCD Status	NCM	Encoder Data Register Empty	Manchester Coder	$FFF2-$FFF3
	NCC	Encoder Completion		
	DCF	Decoder Data Register Full		
	OVF	Overrun Interrupt		
Miscellaneous	KEYF	Keyboard	KB	$FFF0-$FFF1

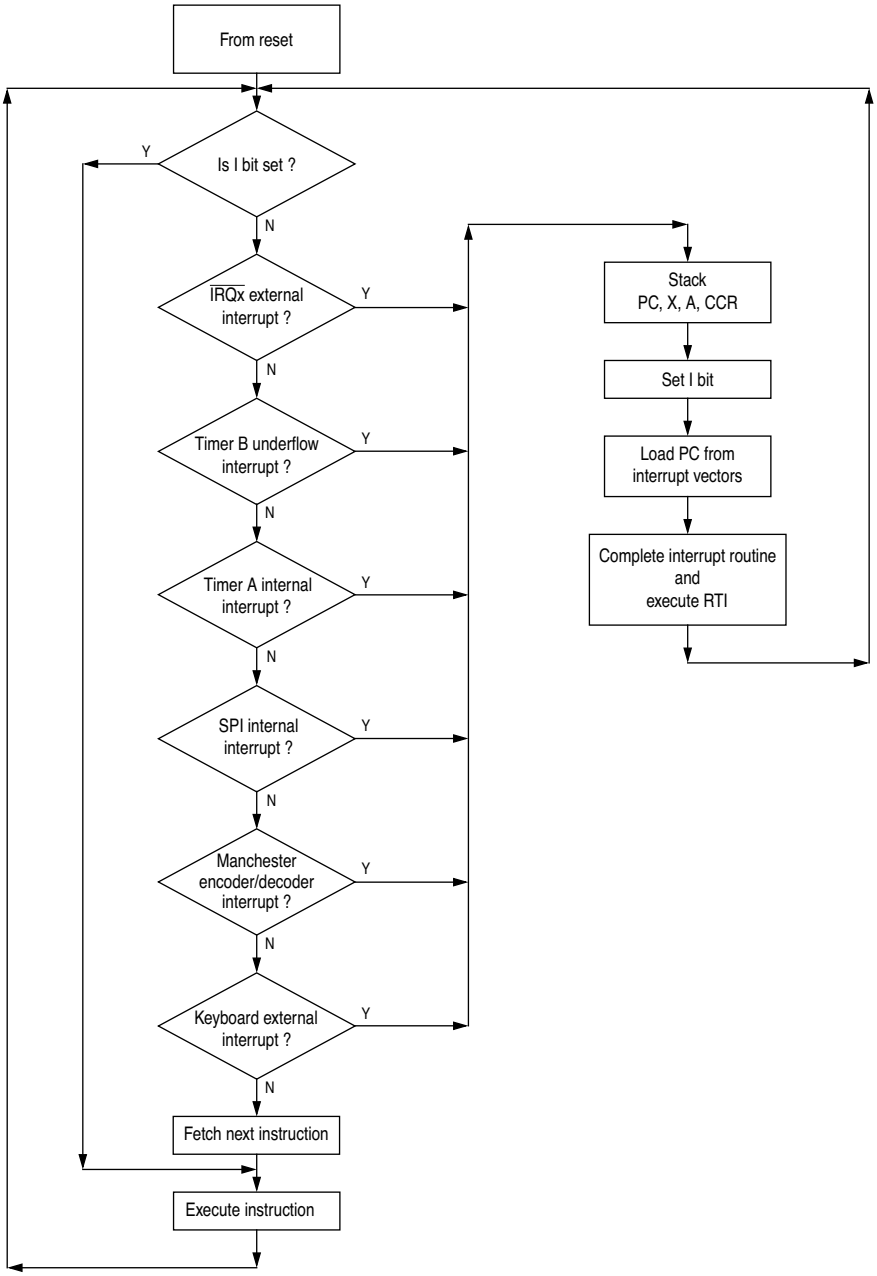


Figure 5-2 Hardware Interrupt Flowchart

5.1 Hardware Controlled Sequences

The following three functions are not strictly interrupts, however, they are tied very closely to the interrupts. These functions are $\overline{\text{RESET}}$, STOP, WAIT.

1) $\overline{\text{RESET}}$ — The $\overline{\text{RESET}}$ input pin causes the program to go to its starting address. This address is specified by the contents of memory locations $FFFE and $FFFF. The interrupt mask of the condition code register is also set. Most parts of the MCU is configured to some known state as described in Table 4-1.

2) STOP — The STOP instruction causes the oscillator to be turned off and the processor "sleeps" until an external interrupt ($\overline{\text{IRQ}}$), keyboard interrupt, or $\overline{\text{RESET}}$ occurs. See Section 11 on Low Power Modes.

3) WAIT — The WAIT instruction causes all processor clocks to stop, but leaves the Timer A, Timer B, MANCD and SPI clocks running. This "rest" state of the processor can be exited by $\overline{\text{RESET}}$, an external interrupt ($\overline{\text{IRQ}}$), keyboard interrupt, Timer or SPI interrupt. There are no special wait vectors for these individual interrupts. See Section 11 on Low Power Modes.

5

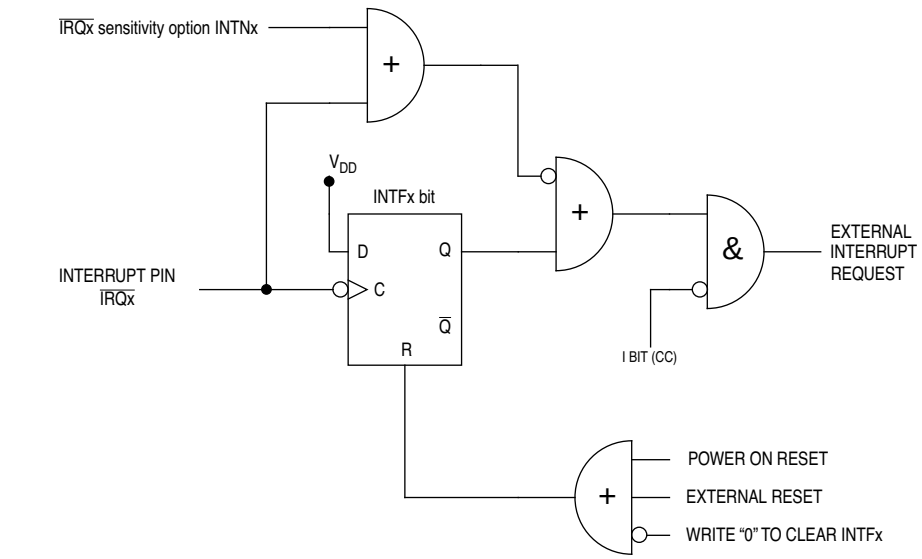
5.2 Software Interrupt (SWI)

The software interrupt is an executable instruction. The action of the SWI instruction is similar to the hardware interrupts. The SWI is executed regardless of the state of the interrupt mask in the condition code register. The service routine address is specified by the contents of memory location $FFFC and $FFFD.

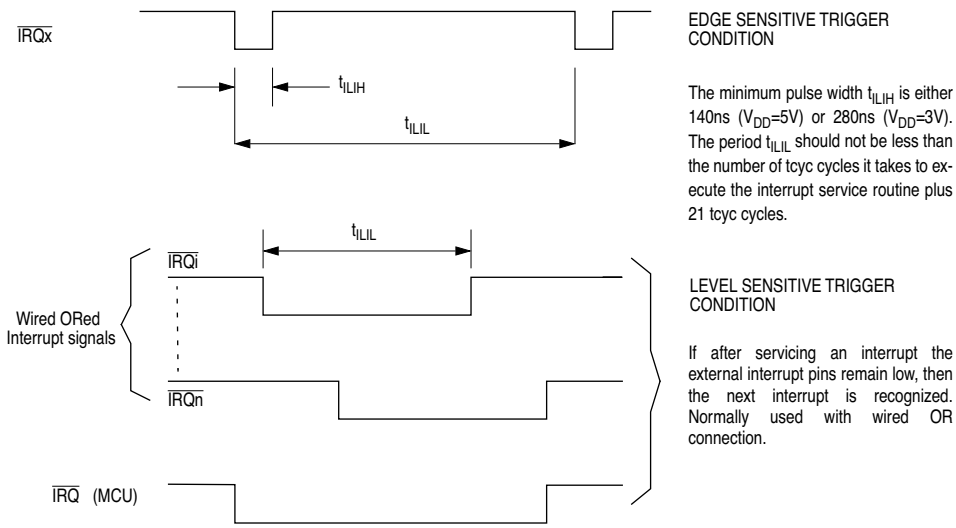
5.3 External Interrupts ($\overline{\text{IRQ1}}$ & $\overline{\text{IRQ2}}$)

The external interrupts $\overline{\text{IRQ1}}$ and $\overline{\text{IRQ2}}$ can be software configured for "negative-edge" or "negative-edge and level" sensitive triggering.

When the signal of the external interrupt pin, $\overline{\text{IRQ1}}$ or $\overline{\text{IRQ2}}$, satisfies the condition selected by the INTN1 and INTN2 bit in the System Option register (bits 4 & 3 of address $35), an external interrupt occurs, and the appropriate INTF flag will be set. The actual processor interrupt is generated only if the interrupt mask bit of the condition code register is also cleared. When the interrupt is recognized, the current state of the processor is pushed onto the stack and the interrupt mask bit in the condition code register is set. This masks further interrupts until the present one is serviced. The service routine address is specified by the contents of $FFFA & $FFFB for both $\overline{\text{IRQ1}}$ & $\overline{\text{IRQ2}}$. After servicing the interrupt, flags are cleared by writing a logic "0" to the corresponding flag; otherwise the CPU will keep servicing the interrupt.



(a) Interrupt Function Diagram



(b) Interrupt Mode Diagram

Figure 5-3 External Interrupt Circuit and Timing

The interrupt logic recognizes negative edge transitions and pulses (special case of negative edges) on the external interrupt lines. Figure 5-3 shows both a block diagram and timing for the interrupt lines ($\overline{IRQ1}$, $\overline{IRQ2}$) to the processor. The first method is used if pulses on the interrupt line are spaced far enough apart to be serviced. The minimum time between pulses is equal to the number of cycles required to execute the interrupt service routine plus 21 cycles. Once a pulse occurs, the next pulse should not occur until the MCU software has exited the routine (an RTI occurs). The second configuration shows several interrupt lines wired-OR to perform the interrupt at the processor. Thus, if the interrupt lines remain low after servicing one interrupt, the next interrupt is recognized.

Note: The internal interrupt latch is cleared in the first part of the service routine; therefore, one (and only one) external interrupt pulse could be latched during t_{ILIL} and serviced as soon as the I bit is cleared.

5.3.1 External Interrupt Triggering Options (INTN1 & INTN2)

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
System Option Register	$35		TCSA1	TCSA0	INTN1	INTN2				-000 0---

INTN1

1 (set) – Negative edge triggering for $\overline{IRQ1}$ only.

0 (clear) – Level and negative edge triggering for $\overline{IRQ1}$.

INTN2

1 (set) – Negative triggering for $\overline{IRQ2}$ only.

0 (clear) – Level and negative edge triggering for $\overline{IRQ2}$.

5.3.2 External Interrupt Enable (INTE1 & INTE2)

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Event Enable Register	$16	TIMHA	INT1E	INT2E						000- ----

INT1E

1 (set) – External interrupt $\overline{IRQ1}$ enabled.

0 (clear) – External interrupt $\overline{IRQ1}$ disabled.

INT2E

1 (set) – External interrupt $\overline{\text{IRQ2}}$ enabled.

0 (clear) – External interrupt $\overline{\text{IRQ2}}$ disabled.

5.3.3 External Interrupt Flags (INTF1 & INTF2)

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Miscellaneous Register	$17	POR	INTF1	INTF2	KEYF					r000 ----

r=1 for POR; r=0 for $\overline{\text{RESET}}$

INTF1

1 (set) – An interrupt on $\overline{\text{IRQ1}}$ pin has occurred.

0 (clear) – An interrupt on $\overline{\text{IRQ1}}$ pin has not occurred.

After servicing this interrupt, this flag should be cleared by writing a “0” to this bit.

INTF2

1 (set) – An interrupt on $\overline{\text{IRQ2}}$ pin has occurred.

0 (clear) – An interrupt on $\overline{\text{IRQ2}}$ pin has not occurred.

After servicing this interrupt, this flag should be cleared by writing a “0” to this bit.

5.4 Keyboard Interrupt

Port pins PA0-PA7 can be configured as keyboard interrupt lines with internal pull-up when the control bits KEYE, KEYX4, KEYX5, KEYX6, and KEYX7 are set. A falling edge of negative pulse with minimum width of T_{ILIH} (250ns) on any configured port A pins will cause a keyboard interrupt to occur. A keyboard interrupt is recognized by the interrupt flag KEYF in the Miscellaneous register (bit 4 of address $17). This flag is cleared by writing a logic “0” to this bit.

When the interrupt is recognized, the current state of the machine is pushed onto the stack and the interrupt mask bit in the condition code register is set. This masks any further interrupt until the present one is serviced. The keyboard interrupt causes the program counter to vector to memory location $FFF0 and $FFF1 which contains the starting address of the interrupt’s service routine.

When configured, the keyboard interrupt lines remain active in during Stop mode. This allows a keyboard interrupt to wake up the MCU when in Stop mode. See Figure 5-4 for keyboard interrupt circuit.

5.4.1 Keyboard Control Register

Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
$34	KEYE				KEYX7	KEYX6	KEYX5	KEYX4	0--- 000

KEYE

1 (set) – PA0-PA3 are configured as keyboard interrupt lines with internal pull-up.

0 (clear) – PA0-PA3 are configured as standard I/O lines.

KEYX7, KEYX6, KEYX5, KEYX4

These four bits configure their corresponding port A lines.

1 (set) – PAx is configured as a keyboard line with internal pull-up.

0 (clear) – PAx is configured as a standard I/O line.

5.5 Programmable Timer (Timer A) Interrupt

Three timer interrupt flags are found in the three most significant bits of the Timer Status register (TSR) at location $19. All three interrupts will vector to the same address at location $FFF6-$FFF7.

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Timer Status Register	$19	ICF	OCF	TOF						uuu- ----

Each flag bit is defined as follows:

TOF - Timer Overflow Flag

TOF is set during the counter transition of $FFFF to $0000. It is cleared by reading the TSR (with TOF set) followed by reading the counter least significant byte ($1F).

OCF - Output Compare Flag

OCF is set when the Output Compare register matches the Counter register. It is cleared by reading the TSR (with OCF set) and then accessing the Output Compare register least significant byte ($1D).

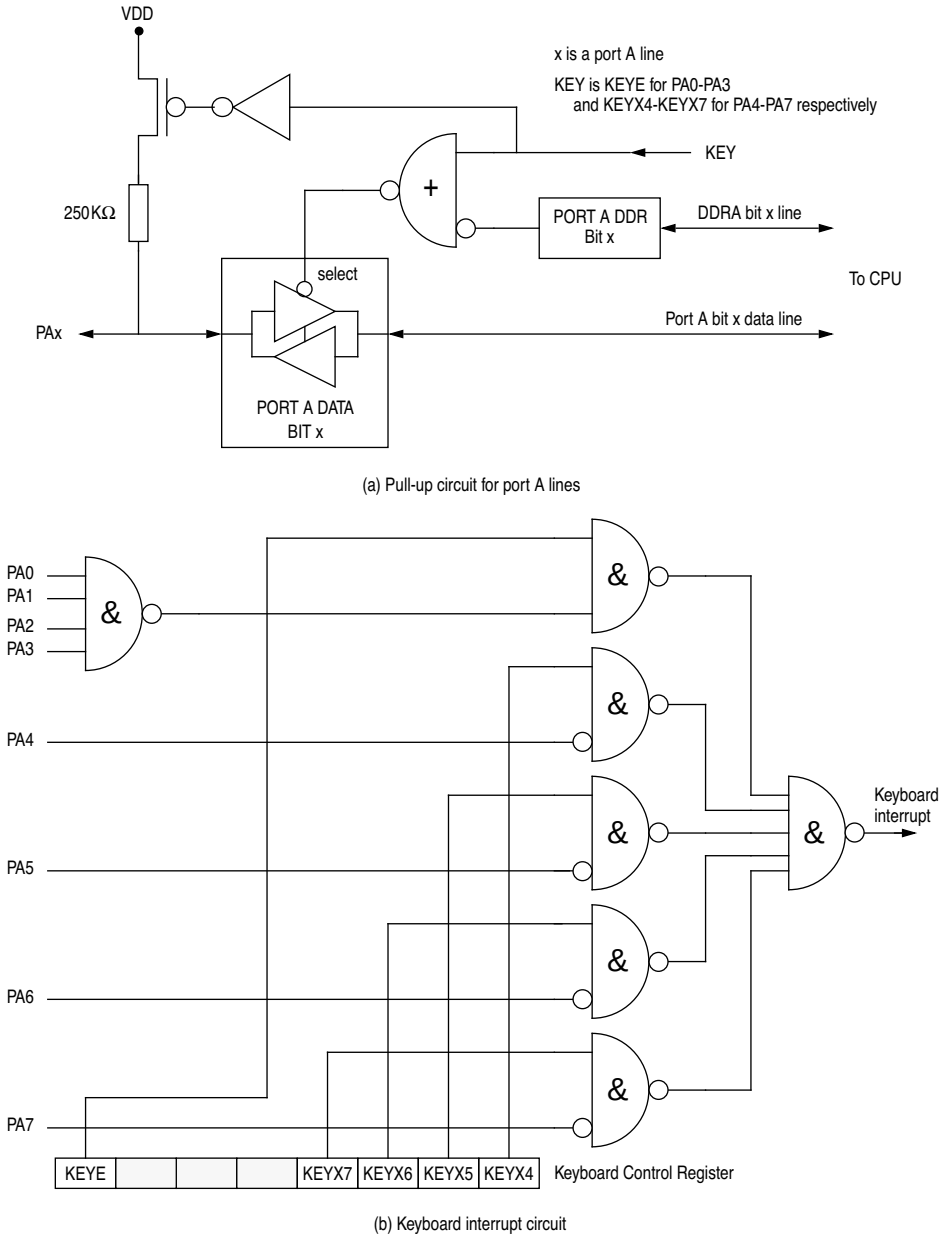


Figure 5-4 Keyboard Interrupt Circuit

ICF - Input Capture Flag

ICF is set when a proper edge has been sensed by the input capture edge detector. It is cleared by an CPU read of the TSR (with ICF set) followed by accessing the Input Capture register least significant byte ($1B).

All three timer interrupt flags have corresponding enable bits (ICIE, OCIE, and TOIE) found in the Timer Control register (TCR) at location $18. Reset clears all enable bits preventing an interrupt from occurring. The actual processor interrupt is generated only if the interrupt mask bit of the condition code register is also cleared. When the interrupt is recognized, the current state of the machine is pushed onto the stack and the interrupt mask bit in the condition code register is set. This masks further interrupts until the present one is serviced. The service routine address is specified by the contents of $FFF6 and $FFF7.

Refer to Section 6.1 for detailed description of Programmable Timer.

5.6 Reloadable Timer (Timer B) Interrupt

Timer B interrupt (TUF) occurs only when the timer B counter rolls over from $0001 to $0000 if the Timer B interrupt enable bit (TBOIE in Timer B Control & Status register $22) is set.

Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
$22	TMBE	TBOIE				TCSB1	TCSB0	TUF	00-- -000

The interrupt service routine address is specified by the contents of memory location $FFF8-$FFF9.

Refer to Section 6.2 - Reloadable Timer B for detailed description.

5.7 SPI Interrupt

An interrupt in the serial peripheral interface (SPI) occurs when the SPI interrupt flag in the Serial Peripheral Status register (bit 7 of address $11) is set, provided the interrupt mask bit in the Condition Code register is cleared and the enable bit in the Serial Peripheral Control register ($10) is enabled. When the interrupt is recognized, the current state of the CPU is pushed onto the stack and the interrupt mask bit in the condition code register is set. This masks any further interrupt until the present one is serviced. The SPI interrupt causes the program counter to vector to memory location $FFF4 and $FFF5 which contains the starting address of the interrupt's service routine. The SPI flag is cleared by accessing the Serial Peripheral Status register (with SPIF set) followed by a read or write of the Serial Peripheral Data register, at location $12.

Refer to Section 7 for detailed description of the Serial Peripheral Interface.

5.8 Manchester Coder (MANCD) Interrupt

A Manchester Coder interrupt occurs when one of the interrupt flags in the MANCD Status register (location $2C) is set, provided the interrupt mask bit in the Condition Code register is cleared and the enable bit in the MANCD Control register ($2B) is enabled. When the interrupt is recognized, the current state of the CPU is pushed onto the stack and the interrupt mask bit in the condition code register is set. This masks any further interrupt until the present one is serviced. The MANCD interrupt causes the program counter to vector to memory location $FFF2 and $FFF3 which contains the starting address of the interrupt's service routine.

Software in the MANCD interrupt service routine must determine the priority and the cause of the MANCD interrupt by examining the interrupt flags located in the MANCD Status register.

There are four interrupt flags associated with MANCD interrupts:

Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
$2C	NCM	NCC	DCF	OVF	-	-	-	-	1100 ----

NCM - Encoder Data Register Empty Flag

1 (set) – Encoder Data register is empty.

0 (clear) – Encoder Data register is not empty.

This bit is cleared by accessing the MANCD Status register (with NCM set), followed by writing to the Encoder Data register.

NCC - Encoding Completion Flag

1 (set) – Encoder is disabled (NCE=0) or, NCE=1 and transmission of the data in the shift register is completed.

0 (clear) – Transmission of data in process.

This bit is cleared by writing to the Encoder Data register when NCE bit is set.

DCF - Decoder Data Register Full Flag

1 (set) – One byte of data received with end pattern verified.

0 (clear) – Decoder Data register not full.

This bit is cleared when the MANCD Status register is accessed (with DCF set) followed by a read of the Decoder Data register, or by clearing the DCE bit.

OVF - Overrun Flag

1 (set) – An overrun has occurred.

0 (clear) – An overrun has not occurred.

This bit is cleared when the DCE bit is cleared.

Refer to Section 8 for detailed description of the Manchester Encoder/Decoder (MANCD).

6
TIMERS

6.1 TIMER A - PROGRAMMABLE TIMER

The timer consists of a 16-bit free-running counter driven by a fixed divide-by-four prescaler. This timer can be used for many purposes, including input waveform measurements while simultaneously generating an output waveform. Pulse widths can vary from several microseconds to many seconds. Figure 6-1 shows a block diagram for the Programmable Timer.

Because the timer has a 16-bit architecture, the I/O registers for the input capture and output compare functions are pairs of 8-bit registers (high byte and low byte). Generally, assessing 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 is also accessed.

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

Ten 8-bit registers are associated with the programmable timer.

- Timer Control Register (TCR) $18
- Timer Status Register (TSR) $19
- Input Capture Register High byte - $1A, Low byte - $1B
- Output Compare Register High byte - $1C, Low byte - $1D
- Counter Register High byte - $1E, Low byte - $1F
- Alternate Counter Register High byte - $20, Low byte - $21

A description of each register is provided in the following paragraphs.

6.1.1 Counter

- Timer A Counter High byte - $1E, Low byte - $1F
- Timer A Alternate Counter High byte - $20, Low byte - $21

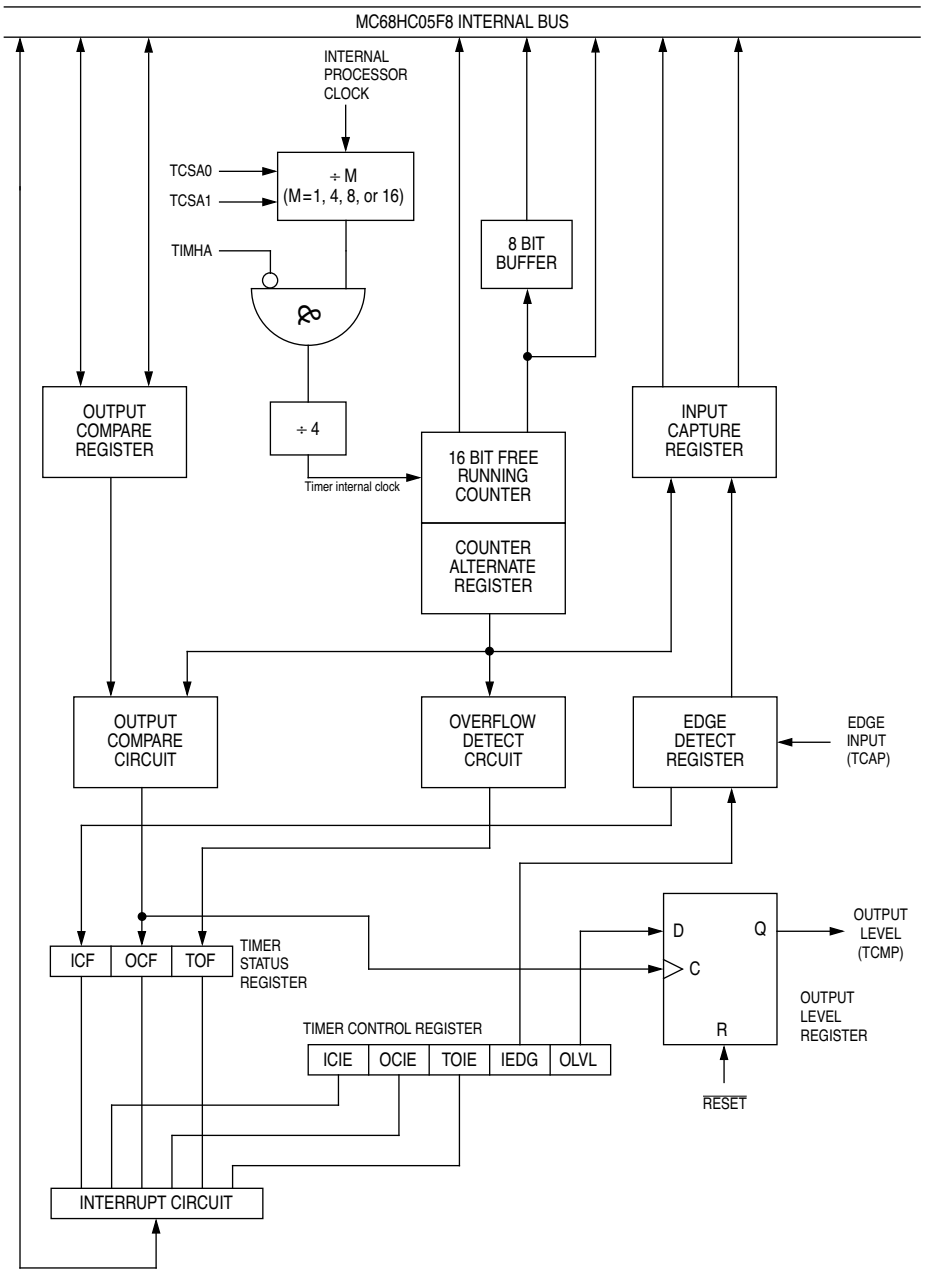


Figure 6-1 Programmable Timer Block Diagram

The key element in the programmable timer is a 16-bit, free-running counter or counter register, preceded by two prescalers. The first stage programmable prescaler provides a slow timer clock by dividing the internal processor clock either by 1, 4, 8, or 16. The second stage is a fixed divide by four prescaler. See Figure 6-1 and Table 6-1 for prescaler values. The counter is incremented during the low portion of the internal bus clock, and the counting can be inhibited by setting the TIMHA bit in the Event Enable register (bit 7 of address $16). Software can read the counter at any time without affecting its value.

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
System Option Register	$35	-	TCSA1	TCSA0	INTN1	INTN2	-	-	-	-000 0---

Table 6-1 Timer A Clock Frequency Selection

TCSA1	TCSA0	Clock Frequency of Timer A
0	0	E/4
0	1	E/16
1	0	E/32
1	1	E/64

Where E = internal bus clock

The double-byte, free-running counter can be read from either of two locations, $1E & $1F (counter register) or $20 & $21 (counter alternate register). Reading only the least significant byte (LSB) of the free-running counter ($1F or $21) receives the count value at the time of the read. If the most significant byte (MSB) ($1E or $20) is read first, the LSB ($1F or $21) is transferred to a buffer. This buffer value remains fixed after the first MSB read, even if the MSB is read several times. This buffer is accessed when the LSB ($1F or $21) is read, and thus, completes a read sequence of the complete counter value.

Reading the timer counter register low byte after reading the timer status register clears the timer overflow flag (TOF), but reading the counter alternate register does not affect TOF. Therefore, the counter alternate register can be read any time without risk of missing timer overflow interrupts due to a cleared TOF.

The free-running counter is preset to $FFFC during reset and is always 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. The value in the free-running counter repeats every ($262144 \div R_{TB}$) internal bus clock cycles (t_{CYC}). R_{TB} is the ratio of timer clock to bus clock frequency, and is dependent on the values of TCSA0 and TCSA1 bits. TOF is set when the counter overflows (from $FFFF to $0000); this will cause an interrupt if TOIE in the Timer Control register is set (bit 5 of address $18).

6.1.2 Output Compare Registers

– Output Compare Register High byte - $1C, Low byte - $1D

The 16-bit Output Compare register is made up of two 8-bit registers. This 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 affected by the timer hardware or reset. If the compare function is not needed, the Output Compare register can be used as storage locations.

The contents of the Output Compare register are continually compared with the contents of the free-running counter and, if a match is found, the Output Compare Flag (OCF) in the Timer Status register is set; and the output level (OLVL) bit is clocked to an Output Level register. The Output Compare register value and the output level bit should be changed after each successful comparison to establish a new elapsed time-out. An interrupt can also accompany a successful output compare provided the interrupt enable bit (OCIE) is set. (The free-running counter is updated every $4 \div R_{TB}$ internal bus clock cycles.)

After a processor write cycle to the output compare register containing the MSB ($1D), the output compare function is inhibited until the LSB ($1D) is also written. The user must write both bytes (locations) if the MSB is written first. A write made only to the LSB ($1D) will not inhibit the compare function. The processor can write to either byte of an output compare register without affecting the other byte. The minimum time required to update the output compare registers is a function of the program rather than the internal hardware. Because the output compare flag and output compare register are not defined at power-on, and not affected by reset, care must be taken when initializing output compare functions with software. The following procedure is recommended:

1) write to Output Compare register high-byte to inhibit further compares;
2) read the Timer Status register to clear OCF;
3) write to Output Compare register low-byte to enable the output compare function.

The output level (OLVL) bit is clocked to the output level register regardless of whether the output compare flag (OCF) is set or clear.

6.1.3 Input Capture Registers

– Input Capture Register High byte - $1A, Low byte - $1B

‘Input Capture’ is a technique whereby an external signal (connected to TCAP pin) is used to trigger a read of the free-running counter. In this way it is possible to relate the timing of an external signal to the internal counter value, and hence to elapsed time.

The two 8-bit registers that 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 valid transition. The level transition that 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 from an input capture will be one greater 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 zero or one count of the free-running counter, which is $4 \times R_{TB}$ internal bus clock cycles.

The free-running counter contents are transferred to the Input Capture register on each valid signal transition 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 MSB ($1A), the counter transfer is inhibited until the LSB ($1B) is also 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 ($1B) does not inhibit the free-running counter transfer since they occur on opposite edges of the internal bus clock.

6.1.4 Timer Control Register (TCR)

Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on Reset
$18	ICIE	OCIE	TOIE	0	0	0	IEDG	OLVL	0000 00u0

The TCR is a read/write register containing five control bits. Three bits control interrupts associated with the three flag bits found in the timer status register (discussed below). The other two bits control: 1) which edge is significant to the input capture edge detector (i.e., negative or positive), and 2) the next value to be clocked to the Output Level registers in response to a successful output compare. The Timer Control register and the free-running counter are the only sections of the timer affected by reset. The TCMP pin is forced low during external reset and stays low until a valid compare changes them to high. Definition of each bit is as follows:

ICIE - Input Capture Interrupt Enable

1 (set) – Input Capture interrupt enabled.

0 (clear) – Input Capture interrupt disabled.

OCIE - Output Compare Interrupt Enable

1 (set) – Output Compare interrupt enabled.

0 (clear) – Output Compare interrupt disabled.

TOIE - Timer Overflow Interrupt Enable

1 (set) – Timer Overflow interrupt enabled.

0 (clear) – Timer Overflow interrupt disabled.

IEDG - Input Edge

1 (set) – TCAP is positive-going edge sensitive.

0 (clear) – TCAP is negative-going edge sensitive.

When IEDG is set, a positive-going edge on the TCAP pin will trigger a transfer of the free-running counter value to the input capture registers. When clear, a negative-going edge triggers the transfer.

OLVL - Output Level Voltage Latch

1 (set) – High output on TCMP pin if counter compare is true.

0 (clear) – Low output on TCMP pin if counter compare is true.

There is a bit in the Event Enable register which may be used to disable and enable the programmable timer.

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Event Enable Register	$16	TIMHA	INTE1	INTE2	0	0	0	0	0	0000 0000

TIMHA - Timer A Enable/Disable

1 (set) – Timer inhibit

0 (clear) – Enable timer (default at reset)

6.1.5 Timer A Status Register (TSR)

Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on Reset
$19	ICF	OCF	TOF	0	0	0	0	0	uuu0 0000

The Timer Status register ($19) contains the status bits for the above three interrupt conditions - ICF, OCF, TOF.

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

ICF - Input Capture Flag

1 (set) – A valid input capture has occurred.

0 (clear) – No input capture has occurred.

This bit is set when the selected polarity of edge is detected by the input capture edge detector; an input capture interrupt will be generated, if ICIE is set, ICF is cleared by reading the TSR and then the Input Capture Low register ($1B)

OCF - Output Compare Flag

1 (set) – A valid output compare has occurred on output compare register.

0 (clear) – No output compare has occurred on output compare register.

OCF will be set when its output compare register contents match that of the free-running counter; an output compare interrupt will be generated, if OCIE is set. OCF is cleared by reading the TSR and then the Output Compare Low register ($1D).

TOF - Timer Overflow Flag

1 (set) – Timer Overflow has occurred.

0 (clear) – No timer overflow has occurred.

This bit is set when the free-running counter overflows from $FFFF to $0000; a timer overflow interrupt will occur, if TOIE (bit 5 in Timer Control register $18) is set. TOF is cleared by reading the TSR and the Counter Low register ($1F).

When using the timer overflow function and reading the free-running counter at random times to measure an elapsed time, a problem may occur whereby the timer overflow flag is unintentionally cleared if:

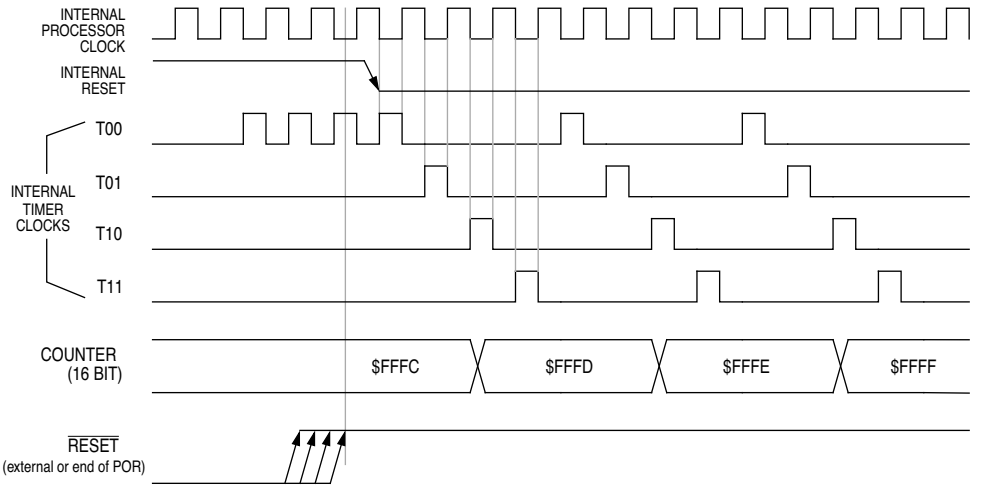
1) the timer status register is read or written when the TOF is set, and
2) the LSB of the free-running counter is read, but not for the purpose of servicing the flag.

Reading the alternate counter register instead of the counter register will avoid this potential problem.

6.1.6 Programmable Timer Timing Diagrams

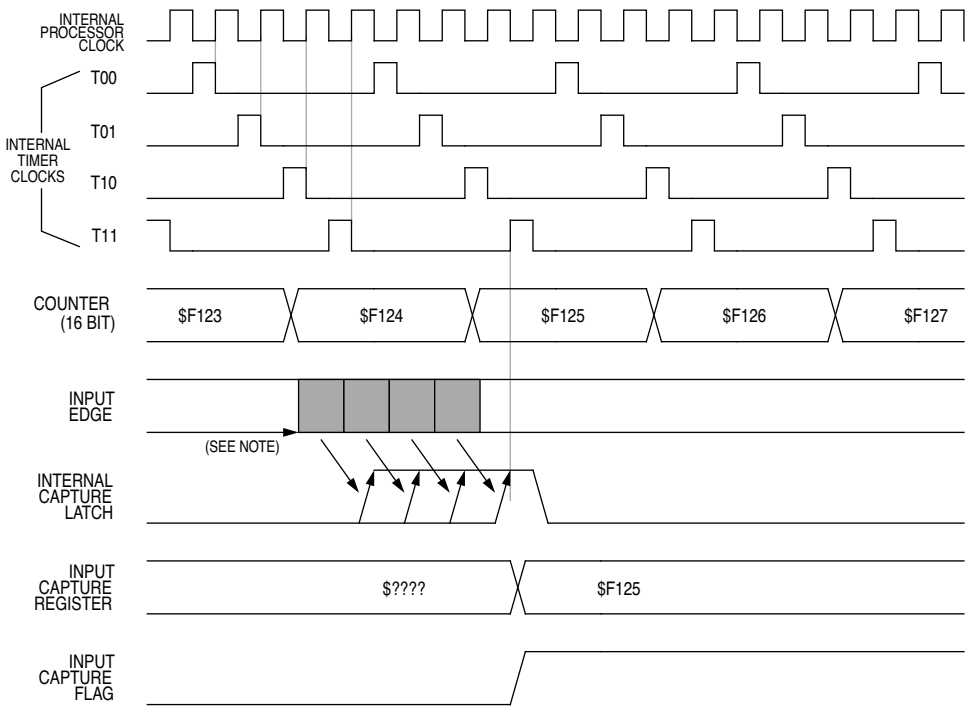
The relationships between the internal clock signals, the counter contents and the status of the flag bits are shown in the following diagrams. It should be noted that the signals labelled ‘internal’ (processor clock, timer clocks and Reset) are not available to the user.

The timing diagrams are for a timer clock frequency of internal bus clock÷4; TCSA0=TCSA1=0.



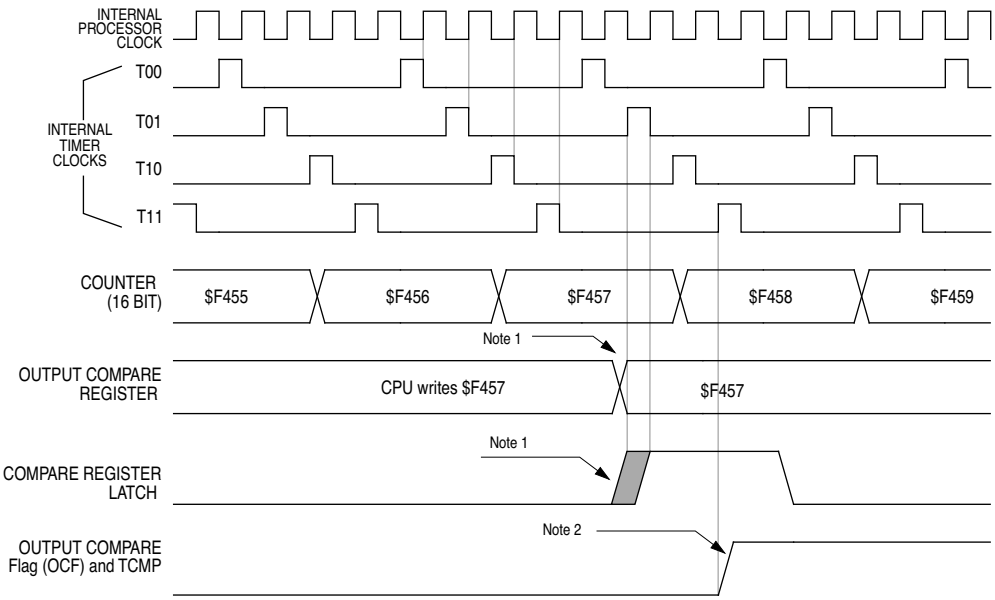
Notes: RESET affects only the Counter register and Timer Control register.

Figure 6-2 Timer State Timing Diagram for Reset



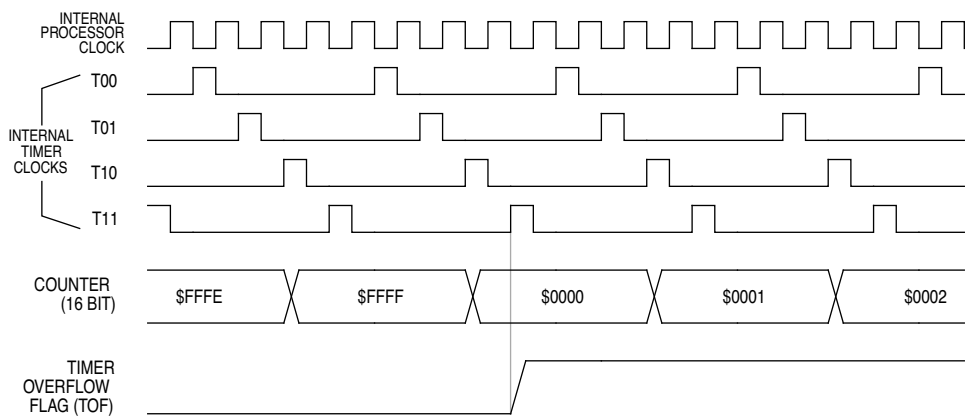
Note: If the input edge occurs in the shaded area from one timer state T10 to the other timer state T10 the input capture flag is set during the next state T11.

Figure 6-3 Timer State Timing Diagram for Input Capture



Note: 1. The CPU write to the compare registers may take place at any time, but a compare only occurs at the timer state T01. Thus a 4-cycle difference may exist between the write to the compare register and the actual compare.
2. The output compare flag is set at the timer state T11 that follows the comparison match ($F547 in this example).

Figure 6-4 Timer State Timing Diagram for Output Compare



Note: The TOF bit is set at timer state T11 (transition of counter from $FFFF to $0000). It is cleared by a read of the timer status register during the internal processor clock high time followed by a read of the counter low register.

Figure 6-5 Timer State Diagram for Timer Overflow

6.2 TIMER B - RELOADABLE TIMER

The Reloadable Timer is similar to the Free-running Timer, with the following differences:

- The 16-bit timer counter is automatically reloaded from the preset timer registers upon underflow occurs.
- There is no input capture function, i.e. no TCAP.
- There is no output compare function, i.e. no TCMP.

The Reloadable Timer is convenient for generating periodic interrupts without the aid of software. In addition, the timer counter value can be read in a similar way to timer A.

6.2.1 Functional Description

See Figure 6-6 for a block diagram of the Reloadable Timer.

The timer B is driven by a clock which is derived from E divided by 4, 8, 16 or 32. In the Control Register, two bits TCSB0 and TCSB1 are used to select the divider for the prescaler, TBOIE is an interrupt enable bit, TMEB is a Timer B enable bit which will inhibit the driving clock when it is clear. Upon reset, the Control Register is cleared, Timer B is disabled, Timer B interrupt is inhibited, the free running counter and the Preset Register are all configured to $FFFF.

The preset register should be written with proper value before enable the Timer B. A low to high transition of the TMBE bit loads the timer counter with the content in the preset register and activates the driving clock, then the free running counter starts to count down, when it rolls over from $0001 to $0000 an interrupt is generated with timer B underflow flag set if the TBOIE is set, meanwhile a “load” signal is produced to reload the counter with the content of the preset register, thus interruption will occur periodically.

6.2.2 Resolution and Maximum Period

When a 3.579MHz crystal is used, the timer resolution and its maximum period are shown in Table 6-2.

Table 6-2 Reloadable Timer Resolution and Maximum Period

TCSB1	TCSB0	FREQUENCY	RESOLUTION	MAX. PERIOD PRESET $FFFF
0	0	E/4	2.25µs	147ms
0	1	E/8	4.5µs	294ms
1	0	E/16	9µs	588ms
1	1	E/32	18µs	1176ms

Where E = internal bus clock

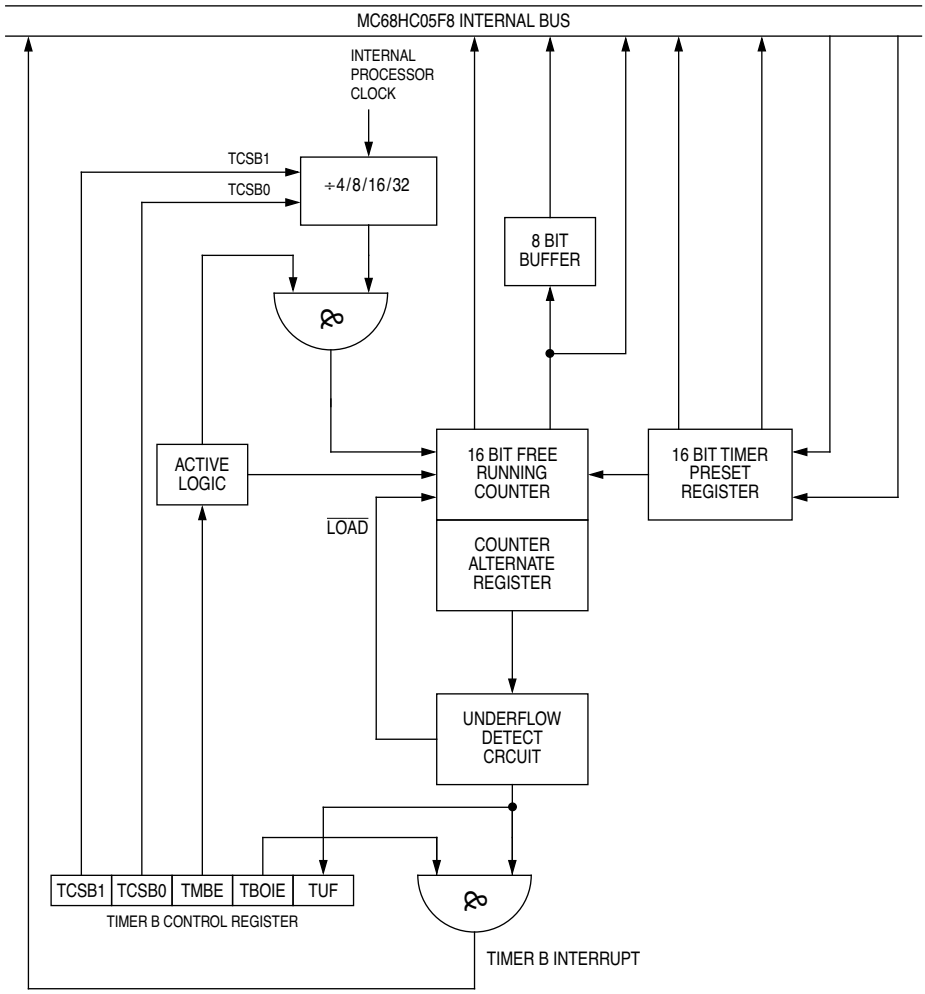


Figure 6-6 Reloadable Timer Block Diagram

6.2.3 Timer B Counter

– Timer B Counter High byte - $25, Low byte - $26
– Timer B Alternate Counter High byte - $27, Low byte - $28

The double-byte, reloadable timer counter can be read from either of two locations, $25 & $26 (Timer B counter register) or $27 & $28 (Timer B counter alternate register). Reading only the least significant byte (LSB) of the free-running counter ($26 or $28) receives the count value at

the time of the read. If the most significant byte (MSB) ($25 or $27) is read first, the LSB ($26 or $28) is transferred to a buffer. This buffer value remains fixed after the first MSB read, even if the MSB is read several times. This buffer is accessed when the LSB ($26 or $28) is read, and thus, completes a read sequence of the complete counter value.

6.2.4 Timer B Preset Register

– Timer B Preset Register High byte - $23, Low byte - $24

On the low to high transition of the TMBE bit, the reloadable timer is loaded with the value set in this 16-bit register.

6.2.5 Timer B Control Status Register

Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
$22	TMBE	TOIE				TCSB1	TCSB0	TUF	00-- -000

TMBE - Timer B Enable/Disable

1 (set) – Timer B enabled.

0 (clear) – Timer B disabled.

Upon reset, this bit is cleared and the driving clock of timer B is inhibited, a low to high transition of this bit loads the timer B counter with the contents of the preset register and activates the driving clock.

TBOIE - Timer B Time-out Interrupt Enable/Disable

1 (set) – Timer B time-out interrupt enabled.

0 (clear) – Timer B time-out interrupt disabled.

When this bit is set, timer B time out interrupt will occur if the time out flag (TUF) is set; otherwise, time out interrupt is disabled.

TCSB1, TCSB0 - Timer B Clock Frequency Select

These two bits are used to select the frequency of timer B driving clock. See Table 6-2.

TUF - Timer B Underflow Flag

This bit is set when the counter of Timer B rolls from $0001 to $0000. It should be cleared by software in the timer B interrupt service routine.

6.3 COP WATCHDOG

A COP (Computer Operating Properly) Watchdog Timer is implemented to restore system operation in the event of system lock-up. This timer consists of a counter which is clocked by a 4 Hz signal; the time-out period is software programmable to approximately 0.5, 1, 2 or 4 seconds (default time out period is 0.5 s after a reset). A watchdog reset occurs when the Watchdog Timer times out, unless the timer is periodically reset by writing to the Watchdog Timer Control Status register.

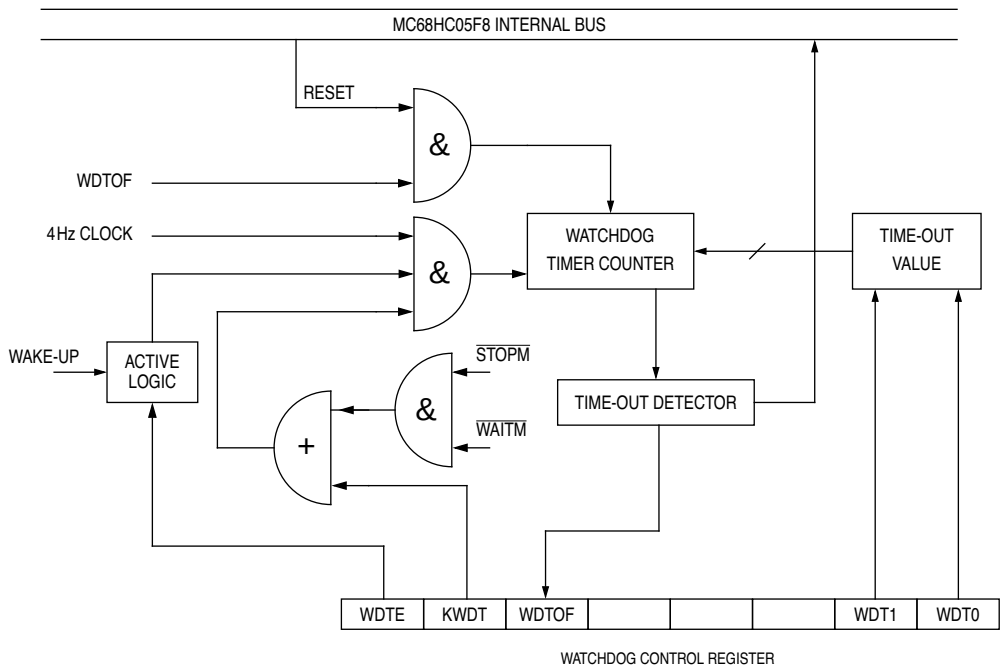


Figure 6-7 Watchdog Timer Block Diagram

6.3.1 Watchdog Timer Time-Out Flag

A watchdog time-out flag (WDTOF) is provided in the Watchdog Control Status register (WDCSR, $36), to allow the user to distinguish between a normal reset (power-on-reset or external reset) and a Watchdog Timer reset. This bit is a logic “1” if the reset was due to a watchdog time-out and logic “0” for a normal reset; and is cleared by reading the WDCSR register. Writing a logic “1” to this bit has no effect on its value other than resetting the watchdog timer counter.

6.3.2 COP System Enable and Operation

A watchdog timer enable (WDTE) bit in the WDCSR is used to enable the COP watchdog system. Its default value is “0” at reset (watchdog disabled).

Writing a “1” to this bit will load the watchdog timer counter with the initial value selected by WDT0 & WDT1 bits and activate the watchdog timer clock. When the watchdog timer counter reaches zero, a watchdog time-out signal is generated to reset the MCU with WDTOF set.

Once the watchdog is enabled, it cannot be disabled by software; writing a “0” to the WDTE bit has no effect.

6.3.3 Disable COP Function in Stop or Wait Mode

A kill watchdog timer (KWDT) bit is provided in the WDCSR to optionally disable and reset the watchdog timer when the STOP or WAIT instruction is executed. This allows the CPU to go into an extended sleep or Wait mode without watchdog timer resets. This feature is not enabled if the KWDT bit is set to “0”.

The KWDT bit permits a “STOP” or “WAIT” instruction to disable the Watchdog Timer. To do so, KWDT must be written to a logic “1” on the first write to the WDCSR after a reset. However, this first write only enables the “kill” feature. A second write of a logic “0” to KWDT must be performed to engage the “kill” feature. After the second write, the execution of a STOP or WAIT instruction will reset the Watchdog Timer and disable the COP watchdog system. Two specific writes are required for this feature to prevent accidental engagement by a single spurious write.

The watchdog counter resumes counting when the MCU exits Stop or Wait mode.

6.3.4 Watchdog Timer Control Status Register (WDCSR)

Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
$36	WDTE	KWDT	WDTOF				WDT1	WDT0	000- --00

WDTE - Watchdog Timer Enable/Disable

1 (set) – Watchdog timer enabled.

0 (clear) – Watchdog timer disabled.

The default for the watchdog timer at reset is disabled. Once enabled by writing a logic “1” to this bit, it cannot be disabled by software. Writing a logic “0” have no effect to this bit.

KWDT - Kill Watchdog Timer Bit

1 (set) – Enable watchdog “kill” feature.

0 (clear) – Disable watchdog “kill” feature.

When this watchdog “kill” feature is set by writing a logic “1” to this bit immediately after a reset, the watchdog timer will be disabled when the MCU in Stop or Wait mode. The default for the watchdog timer “kill” feature at reset is disabled.

Reading this bit will show the value of first write after Reset or the default value upon reset.

WDTOF - Watchdog Timer Time-out Flag

1 (set) – A watchdog timer time-out has occurred.

0 (clear) – A watchdog timer time-out has not occurred.

This flag is cleared by writing a logic “0” to this bit. Writing a logic “1” to this bit will reset the watchdog timer counter, and hence avoiding a watchdog time-out. The write does not affect the time-out flag.

WDT1, WDT0 - Time-out Period Select

WDT1	WDT0	Time-Out Period	
		Min. (s)	Max. (s)
0	0	0.25	0.5
0	1	0.75	1.0
1	0	1.75	2.0
1	1	3.75	4.0

THIS PAGE LEFT BLANK INTENTIONALLY

7
SERIAL PERIPHERAL INTERFACE

The serial peripheral interface (SPI) is an interface built into the MC68HC05F8 MCU which allows two MC68HC05F8 MCUs to be interconnected within a single “black box” or on the same printed circuit board. In a serial peripheral interface (SPI), separate wires (signals) are required for data and clock. In the SPI format, the clock is not included in the data stream and must be furnished as a separate signal. An SPI system may be configured in one containing one master and one slave MCUs.

Figure 7-1 illustrates a normal system configurations. In this system three basic lines (signals) are required for the Serial Clock (SCK), Serial Data Input (SDI), and Serial Data Output (SDO) lines.

7.1 Features

- Full duplex, three-wire synchronous transfer
- Master or slave operation
- 447.5 KHz master bit frequency
- 1.79 MHz (Max.) slave bit frequency
- End of transmission interrupt flag
- Data collision flag protection

7.2 Signal Description

The three basic signals (SCK, SDO and SDI) are described in the following paragraphs. Each signal function is described for both the master and slave mode. Figure 7-2 summarizes the SPI port timing for data exchange operation.

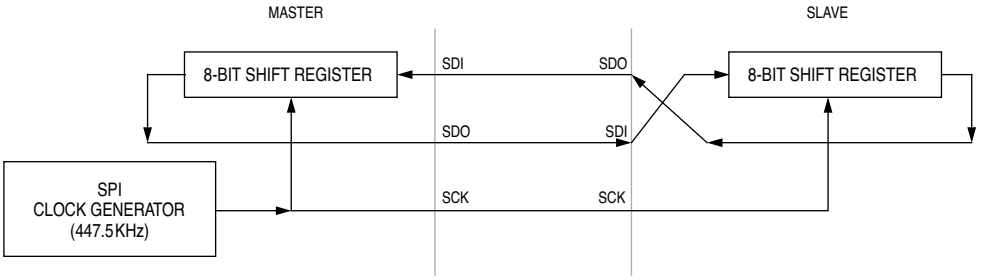


Figure 7-1 SPI Master-Slave Interconnection

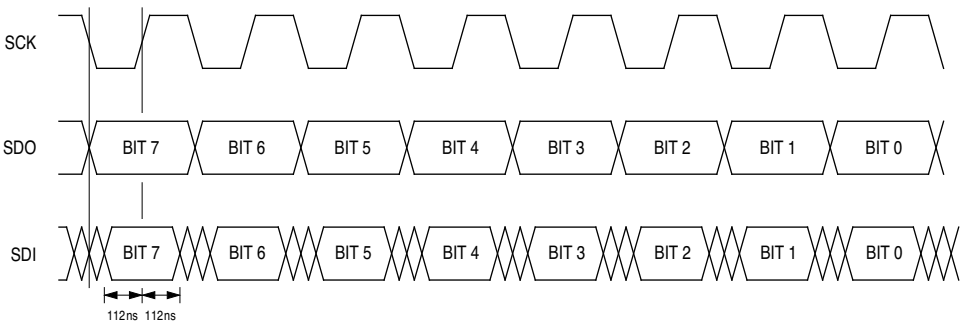


Figure 7-2 SPI Port Timing

7.2.1 Serial Clock (SCK)

The state of SCK between transmissions must be logic “1”. The first falling edge of SCK signals the beginning of a transmission. At this time the MSB bit of received data is accepted at the SDI pin and the MSB bit of transmitted data is presented at the SDO pin. Data is captured at the SDI pin on the rising edge of SCK. Subsequent falling edges shift the data, and accept the next received data bit at SDI pin, and present the next transmitted data bit at SDO pin. The transmission is ended upon the receipt of the LSB bit.

In Master Mode, the format is identical except that the SCK pin is an output and the shift clock now originates internally. The Master Mode transmission frequency is fixed at E/4.

Care should be taken when enabling the SPI; additional clock edges may be present when the port is switched from standard I/O to SPI.

7.2.2 Serial Data Output (SDO)

Data is transmitted in MSB first format. The state of the SDO pin will always reflect the value of the first bit receive on the previous transmission if there was one. Prior to enabling the SPI, PD5 can be initialized to determine the beginning state if a standard output since that pin is coupled to the last stage of the serial shift register. On the first falling edge of SCK the first data bit to be shifted out is presented to the output pin.

7.2.3 Serial Data Input (SDI)

The SDI pin becomes an input as soon as the SPI pin on the falling edge of SCK. Valid data must be present at least 112ns before the rising edge of the clock and remain valid for 112ns after the edge.

7.3 General Operation

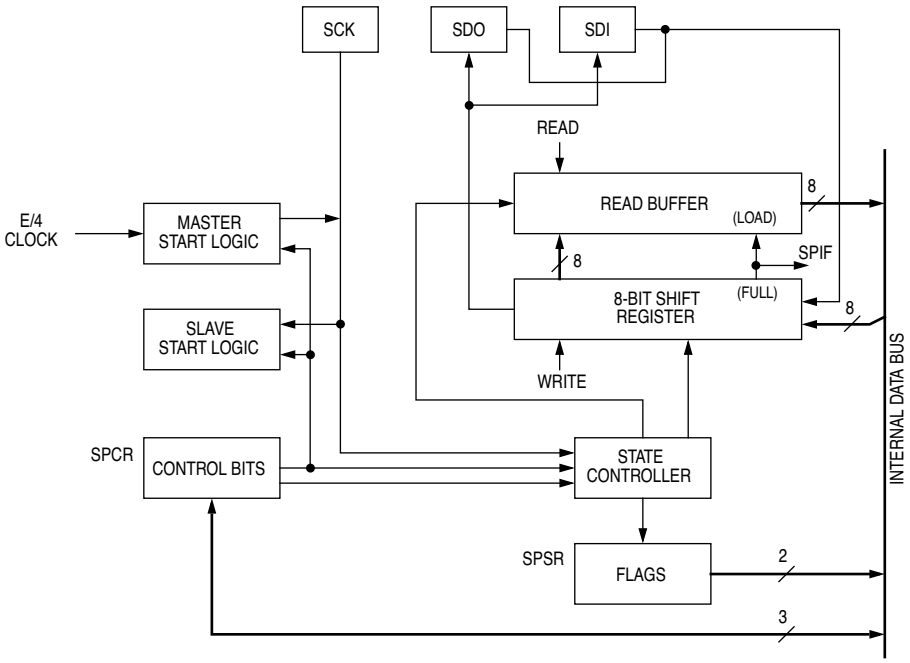
A block diagram of the serial peripheral interface (SPI) is shown in Figure 7-3. In a master configuration, the master start logic originates the system clock (SCK) based on the 447.5KHz (or the E/4) clock. This clock is also used internally to control the state controller as well as the 8-bit shift register. As a master device, data is parallel loaded into the 8-bit shift register (from the internal bus) during a write cycle, data is applied serially from a slave device via SDI pin to the 8-bit shift register. After the 8-bit shift register is loaded, its data is parallel transferred to the read buffer and then is made available to the internal data bus a CPU read cycle.

In a slave configuration, the slave start logic receives a system clock input (from the master device) at SCK pin. Thus, the slave is synchronized with the master. Data from the master is received serially at the slave SDI and loads the 8-bit shift register. After the 8-bit shift register is loaded, its data is parallel transferred to the read buffer and then is made available to the internal data bus during a CPU read cycle. During a write cycle, data is parallel loaded into the 8-bit shift register from the internal data bus and then shifted out serially to the SDO pin for application to the master device.

One point to be noted, the SCK pin needs to be externally pulled high with 10K Ohms, in order to bias the initial states at logic high.

7.4 SPI Registers

There are three registers associated with the serial parallel interface. They are the Serial Peripheral Control register (SPCR, location $10), the Serial Peripheral Status register (SPSR, location $11), and the Serial Peripheral Data I/O register (SPDR, location $12). Each register are described below.



Note: SCK, MOSI, and MISO are external pins; where

SCK - provides system clock when device is configured as a master unit. Receives system clock when device is configured as a slave unit.

SDI - provides serial output to slave unit when device is configured as a master. Receives serial input from master when device is configured as a slave unit.

SDO - receives serial input from slave unit when device is configured as a master. Provides serial output to master when device is configured as a slave unit.

Figure 7-3 SPI Block Diagram

7.4.1 SPI Control Register (SPCR)

Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
$10	SPIE	SPE		MSTR					00-0 ----

SPIE - Serial Peripheral Interrupt Enable

When the serial peripheral interrupt enable bit is high, it allows the occurrence of a processor interrupt and forces the proper vector to be loaded into the program counter if the serial peripheral

status register bit (SPIF) is set to a logic one. It does not inhibit the setting of a status bit. The SPIE bit is cleared by reset.

SPE - Serial Peripheral Enable

When set, this bit enables the Serial I/O Port and initializes the Port D DDR such that PD5 (SDO) is output, PD6 (SDI) is input and PD7 (SCK) is input (Slave Mode only). The Port D DDR can be subsequently altered as the application requires and the Port D data register (except for PD5) can be manipulated as usual, however these actions could affect the transmitted or received data. When SPE is cleared, Port D reverts to standard parallel I/O without affecting the Port D data register or DDR. SPE can be read or written any time, but clearing SPE while a transmission is in progress will abort the transmission, reset the bit count and return Port D to its normal I/O function. Reset clears this bit.

MSTR - Master Bit

When set, this bit configures the SPI for Master Mode. This means that the transmission is initiated by a write to the data register and the SCK pin becomes an output providing a synchronous data clock at a fixed rate of E clock divided by 4. While the device is in Master Mode, the SDO and SDI pins do not change function. These pins behave exactly as they would in Slave Mode. Reset clears this bit and configures the SPI for Slave operation. MSTR may be set at any time regardless of the state of SPE. Clearing MSTR will abort any transmission in progress.

7.4.2 SPI Status Register (SPSR)

Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
$11	SPIF	DCOL							00-- ----

SPIF - Serial Peripheral Interface Flag

The serial peripheral data transfer flag bit notifies the user that a data transfer between the device and an external device has been completed. With the completion of the data transfer, SPIF is set, and if SPIE is set, a serial peripheral interrupt (SPI) is generated. During the clock cycle that SPIF is being set, a copy of the received data byte in the shift register is moved to a buffer. When the data register is read, it is the buffer that is read.

The transfer of data is initiated by the master device writing its serial peripheral data register.

Clearing the SPIF bit is accomplished by a software sequence of accessing the serial peripheral status register while SPIF is set and followed by a write to or a read of the serial peripheral data register. While SPIF is set, all writes to the serial peripheral data register are inhibited until the serial peripheral status register is read. This occurs in the master device. In the slave device, SPIF can be cleared before the second SPIF in order to prevent an overrun condition. The SPIF bit is cleared by reset.

DCOL - Data Collision

This is a read only status bit which indicates that an invalid access to the data register has been made. This can occur any time after the first falling edge of SCK and before SPIF is set. A read or write of the data register during this time will result in invalid data being transmitted or received.

DCOL is cleared by reading the status register with SPIF set followed by a read or write of the data register. If the last part of the clearing sequence is done after another transmission has been started, DCOL will be set again. Reset also clears this bit.

7.4.3 Serial Peripheral Data Register (SPDR)

Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
$12								

The serial peripheral data I/O register is used to transmit and receive data on the serial bus. Only a write to this register will initiate transmission/reception of another byte and this will only occur in the master device. A slave device writing to its data I/O register will not initiate a transmission. At the completion of transmitting a byte of data, the SPIF status bit is set in both the master and slave devices. A write or read of the serial peripheral data I/O register, after accessing the serial peripheral status register with SPIF set, will clear SPIF.

During the clock cycle that the SPIF bit is being set, a copy of the received data byte in the shift register is being moved to a buffer. When the user reads the serial peripheral data I/O register, the buffer is actually being read. During an overrun condition, when the master device has sent several bytes of data and the slave device has not internally responded to clear the first SPIF, only the first byte is contained in the receive buffer at any time. The first SPIF must be cleared by the time a second transfer of data from the shift register to the read buffer is initiated or an overrun condition will exist.

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

The ability to access the serial peripheral data I/O register is limited when a transmission is taking place. It is important to read the discussion defining the DCOL and SPIF status bits to understand the limits on using the serial peripheral data I/O register.

8
MANCHESTER ENCODER/DECODER

The built-in full duplex Manchester Coder (MANCD) performs data format conversion between parallel NRZ and serial Manchester code. The bit rate is programmable, in four steps between 600 and 4800 baud when a 3.579MHz crystal is used.

Data are encoded to Manchester codes by writing to the Encode register before it is output to the ENCOOUT pin. The data transfer format is 2 sync bits followed by 8 data bits and a trailing bit. Two low level bits are used as a pause between byte transfers. Figure 8-3 shows the one byte data transfer.

Manchester codes enter the MCU, LSB first, at the DECOIN pin to the Decode register, gets decoded before processing. The idle state of the DECOIN pin is a logic high. A Schmitt trigger is built in at the DECOIN input to improve noise immunity.

8.1 Features

- Four programmable bit rates
- Buffered encode data register and decode data register
- Encode data register empty flag and interrupt
- Encoding complete flag
- Decode data register full flag and interrupt
- Decode overrun flag and interrupt
- Bit format error detection
- Bit rate error detection
- Built-in front end Schmitt trigger in decoder

8.2 General Operation

Figure 8-1 shows a block diagram of the Manchester encoder/decoder. Logic flow of the hardware operation of the encoder and decoder are shown in Figure 8-2 and Figure 8-3 respectively.

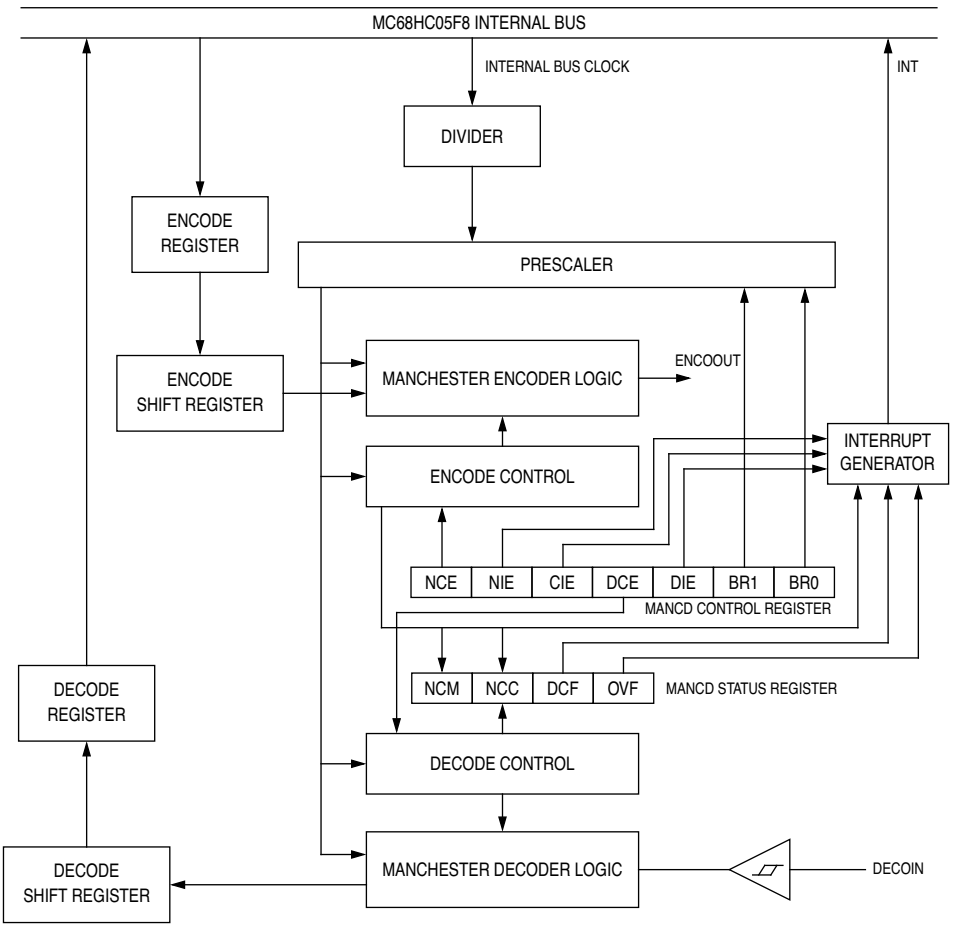


Figure 8-1 Manchester Encoder/Decoder Block Diagram

8.2.1 Encoder

The Manchester Encoder is used to convert data from NRZ format to Manchester Code format; and output onto the ENCOOUT pin.

8.2.1.1 Idle State of Encoder

Upon reset the encoder enable bit (NCE) is cleared, ENCOOUT pin is at high impedance, internal encoding clock is inhibited, and the encoder is in the idle state. The encode data register empty flag (NCM) and the encoding completion flag (NCC) in the status register are both set.

8.2.1.2 Initialization of Encoder

The encoder is initialized by configuring the bit rate control bits (BR0, BS1) and setting NCE=1 to place the encoder in the standby state. The encoding process is initiated by writing to the Encoder Data register, which is then transferred to the encode data shift register ready for encoding. After 2 delay bits (ENCOOUT pin is low) and 2 sync bits, the encoded data is shifted out to the ENCOOUT pin, LSB first. See Figure 8-3 for a graphical representation.

8.2.1.3 Encode Data Register Empty Flag (NCM) and Encode Interrupt

After the last data bit in the encode data shift register is encoded and output to ENCOOUT, a trailing bit followed by two pause bits are generated to conclude a one byte transmission. After this, if the Encode Data register is not empty, the encoding process is repeated.

When data from the encode data register is transferred to the encode data shift register, the encoder data register empty flag (NCM) is set, causing an interrupt to be generated if the encode interrupt is enabled (i.e. NIE = 1). The next byte of data to be encoded can be written in to the encoder data register in an interrupt service routine. The NCM bit is automatically cleared by writing to the encode data register after accessing the MANCD Status register.

8.2.1.4 End Pattern Generation and Next Data Byte Encoding

The end pattern of one byte sequence is generated automatically after the last bit. This pattern consists of a trailing bit and two pause bits. After this, if the Encode Data register is empty, ENCOOUT is set to high impedance, and the encoder returns to the standby state. The encoding complete flag (NCC) will be set, and an interrupt is generated if the encoding complete interrupt enable bit (CIE) is set. If the encode data register is not empty, the next encoding is started.

8.2.1.5 Disable Encoder

The encoder is disabled by setting NCE=0; causing the ENCOOUT pin to be tri-stated. If the NCE bit is cleared while an encoding is in progress (indicated by NCC=0), the encoder will complete encoding of the current byte, plus the end patterns, before going into idle.

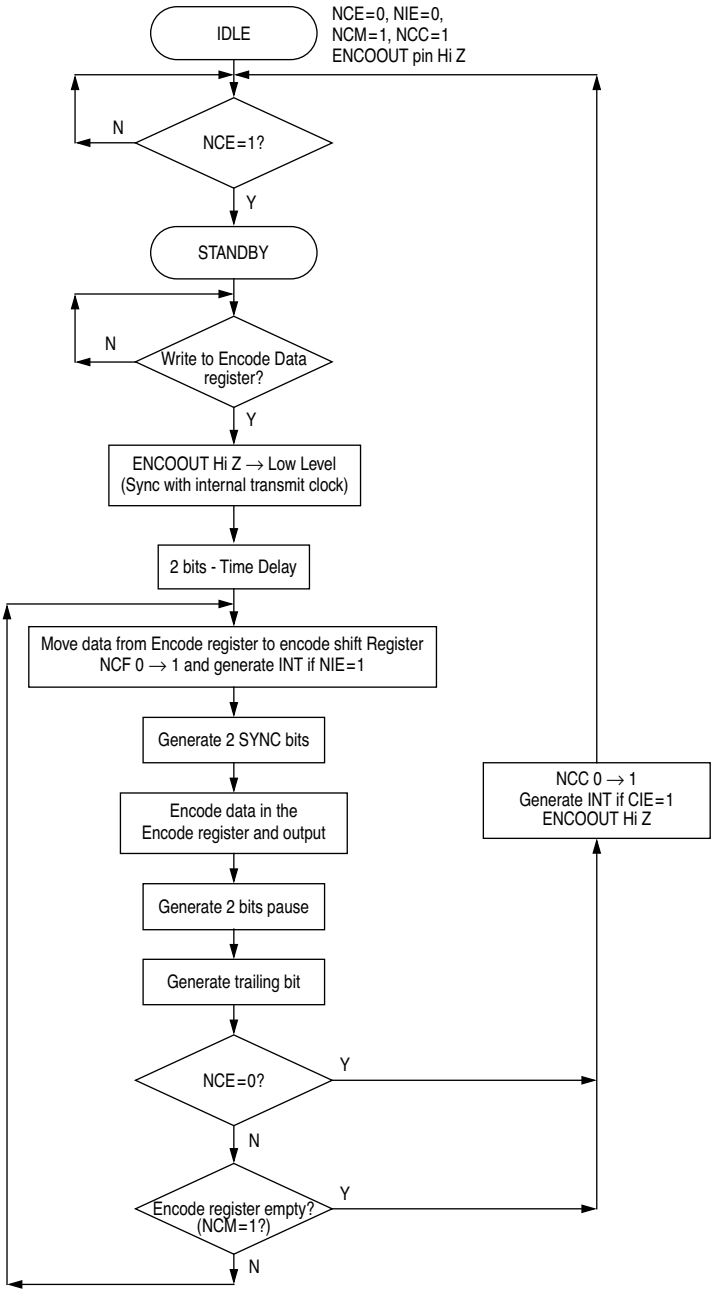


Figure 8-2 Logic Flow of Encoder Hardware Operation

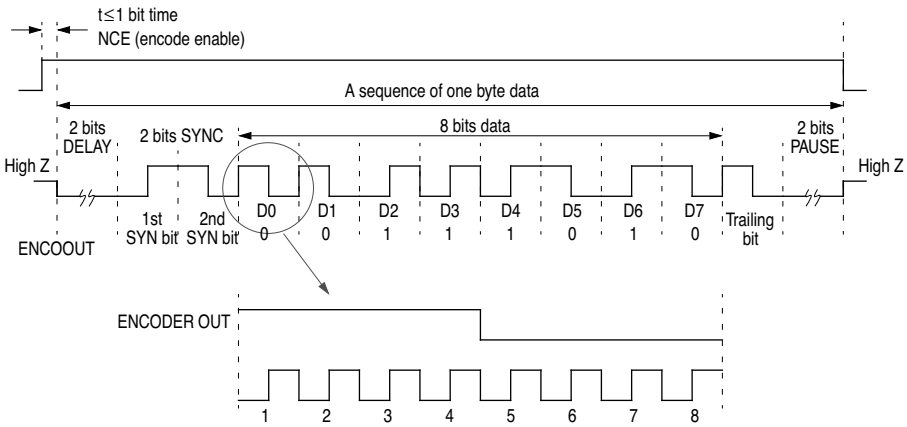


Figure 8-3 Encoder Timing Diagram

8.2.2 Decoder

The Manchester decoder is used to convert incoming Manchester codes on the DECOIN pin to NRZ data format for processing.

Upon reset the decoder is disabled, decoder enable bit DCE=0. To initiate the decoding process, the bit rate is first configured. Setting DCE activates the internal decoding clock, the decoder enters the start state and the DECOIN pin begins to be sampled. After a low state is confirmed, the receiver starts to hunt for the 2 bits SYNC pattern. if it is detected, the decoding procedure starts, the decode logic converts the data bits from Manchester code format to NRZ format and shifts the result to the decode shift register bit by bit. After all 8 bits have been received and converted to one data byte, the end pattern of a trailing bit plus two bit pause is verified. If the pattern followed is correct, the decode flag is set and an interrupt is generated, otherwise the decoder is reset and returns to the start state.

8.2.2.1 Decoder Overrun

After one byte of data is received and end pattern verified, the decode output flag (DCF) is checked first, if it is zero (indicating the Decode Register is empty), one byte of data which has been received is loaded to the Decode Register and interrupt is generated with the decode output flag set (DCF=1), otherwise the receive overrun flag is set and an interrupt is generated.

8.2.2.2 Data Bit Format Error Detection

During decoding, a bit format error detection is performed. If 00 or 11 appears at a time interval in which one bit of data is expected, which means that bit format error occurs, then the decoder is reset and returns to the start state.

8.2.2.3 Bit Rate Error Detection

During decoding, the input data is sampled by an internal clock, of which the frequency is 8 times of the selected bit rate. If the bit rate of the input data varies exceeding 10% with reference to the nominal value (see bit rate selection table), a bit rate error occurs and the data which is being received is discarded. In this case the decoder is initialized and returns to the start state.

8.3 Manchester Encoder/Decoder Registers

8.3.1 MANCD Control Register

Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
$2B	NCE	NIE	CIE	DCE	DIE		BR1	BR0	0000 0-00

NCE - Encoder Enable Bit

1 (set) – Enable the encoder. A transition from 0 to 1 of this bit initiates transmission sequence of one byte data, including 2 proceeding idle bits and 2 ending pause bits.

0 (clear) – Disable the encoder. When this bit is cleared, the encoder (except the control bits) is reset and put in idle state.

If the NCE bit is cleared while an encoding is in progress (indicated by NCC=0), the encoder will complete encoding of the current byte, plus the end patterns, before going idle. Clearing and setting the NCE bit during encoding of a byte has no effect on the encoder operation. Normally, after the last byte of data is written to the Encode Data register, the NCM bit will generate an interrupt (if NIE=1), indicating that data have been transferred to the encode data shift register. The user should then clear the NCE bit to put the encoder in the idle state.

NIE - Encoder Interrupt Enable Bit

1 (set) – Enable the encoder interrupt. If this bit is set, interrupt is generated when the NCM flag is set.

0 (clear) – Disable the encoder interrupt.

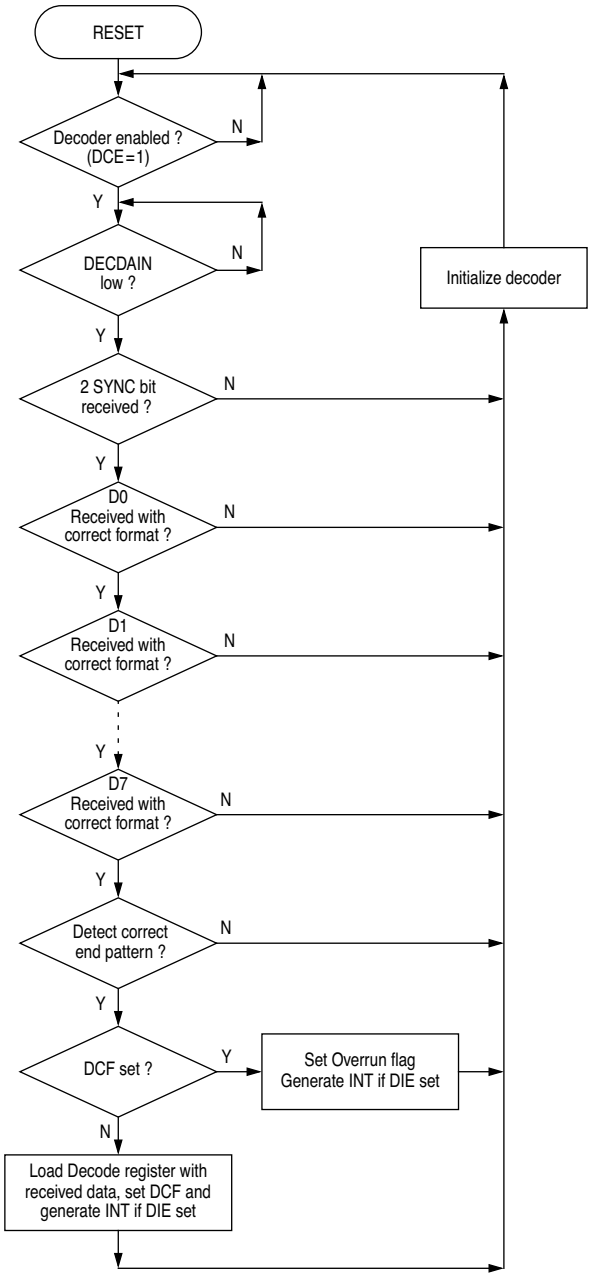


Figure 8-4 Logic Flow of Decoder Hardware Operation

CIE - Encoding Complete Interrupt Enable Bit

1 (set) – Enable encoding complete interrupt. If this bit is set, interrupt is generated when the NCC flag is set.

0 (clear) – Disable the encoding complete interrupt.

DCE - Decoder Enable

1 (set) – Enable the decoder.

0 (clear) – Disable the decoder. When this bit is cleared, the decoder is reset, receive and decode function is disabled.

DIE - Decode Interrupt Enable

1 (set) – Enable the decoder interrupt. If this bit is set, interrupt is generated when the DCF or OVF flag is set.

0 (clear) – Disable the decoder interrupt.

BR1 & BR0 - Bit Rate Select

These two bits are used to select the transfer bit rate.

BR1	BR0	Bit Cycle	Bit rate (3.579 MHz crystal)
0	0	1/8 (E/372)	601
0	1	1/4 (E/372)	1203
1	0	1/2 (E/372)	2405
1	1	E/372	4810

"bit" refers to bit unit in NRZ format, i.e. one bit is twice the bit unit in Manchester format. E = internal bus clock

8.3.2 MANCD Status Register

Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
$2C	NCM	NCC	DCF	OVF					1100 ----

NCM - Encoder Data Register Empty Flag

The Encoder Data register empty flag is set to indicate the contents of the Encoder Data register have been transferred to the encode data shift register. If the NCM bit is clear, it indicates that the transfer has not yet occurred and a write to the Encode Data register will overwrite the previous value. This bit is cleared by accessing the MANCD status register (with NCM set), followed by writing to the Encode Data register. Reset sets the NCM bit.

NCC - Encoding Completion Flag

This bit is set to indicate that no data transmitting or encoding is in progress. It is set when one of the following cases occurs:

1) The encoder is disabled, i.e. NCE=0, transmission of the data in the encode data shift register is completed.
2) The encoder is enabled, NCE=1, the encode data register is empty (NCM=1) and transmission of the data in the encode data shift register is completed.

Writing to the Encoder Data register when the NCE bit is set clears this flag. Reset or clearing the NCE bit sets this NCC bit.

DCF - Decoder Data Register Full Flag

This bit is set when one byte of data is received with end pattern verified, and an interrupt is generated if the decoder interrupt is enabled (DIE=1). This flag is cleared when the Status register is accessed (with DCF set) followed by a read of the Decode Data register, or by clearing the DCE bit.

OVF - Overrun Flag

When an overrun occurs, this flag is set, and an interrupt is generated if the decode interrupt is enabled. Clearing the DCE bit will reset the decoder and thus clearing this flag. See Section 8.2.2.1 for definition of an overrun condition.

8.3.3 Encode Data Register ($2D)

Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
$2D								

This is a write only register. Data written to this register will be encoded to Manchester format and then transmitted out to the ENCOOUT pin in sequential format.

8.3.4 Decode Data Register ($2E)

Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
$2E								

The is a read only register. Data in Manchester format entering the DECOIN pin will be decoded and the result placed in this register.

THIS PAGE LEFT BLANK INTENTIONALLY

9
DTMF/MELODY GENERATOR

The DTMF/Melody Generator (DMG) is a multi-function tone generator built into the MC68HC05F8 MCU, supporting DTMF dialling, melody-on-hold, and pacifier tone functions. The associated output pins are TONEOUT and TONEX.

9.1 Features

- 4 row and 4 column frequencies for DTMF dialling
- 24 row and 24 column frequencies for dual tone melody
- 28 frequencies for pacifier tone to acknowledge button pressed for pulse dialling
- Power saving mechanism for no tone condition
- 3.579MHz ÷ 2 operation
- 6-bit D/A converter and 28 time steps for sine wave generation
- Sine wave or square wave selectable output for melody or DTMF
- Single or dual tone capability for melody or DTMF

9.2 General Operation

In Figure 9-1, the DMG consists of a row tone and a column tone generation path. The tone frequency of each path is controlled by their respective frequency control registers; Row Frequency Control register (FCR) and Column Frequency Control register (FCC). At the TONEOUT output, single/dual sine/square wave tones of DTMF and melody frequencies are possible, whereas at the TONEX output, only single square wave tones are possible.

To generate a sine wave tone with programmable frequency in a path, the internal clock (i.e. the 3.58MHz ÷ 2) is first divided by a frequency divider, whose value is set by the frequency control register (FCR or FCC). The output of the divider is a periodic pulse train whose frequency is the

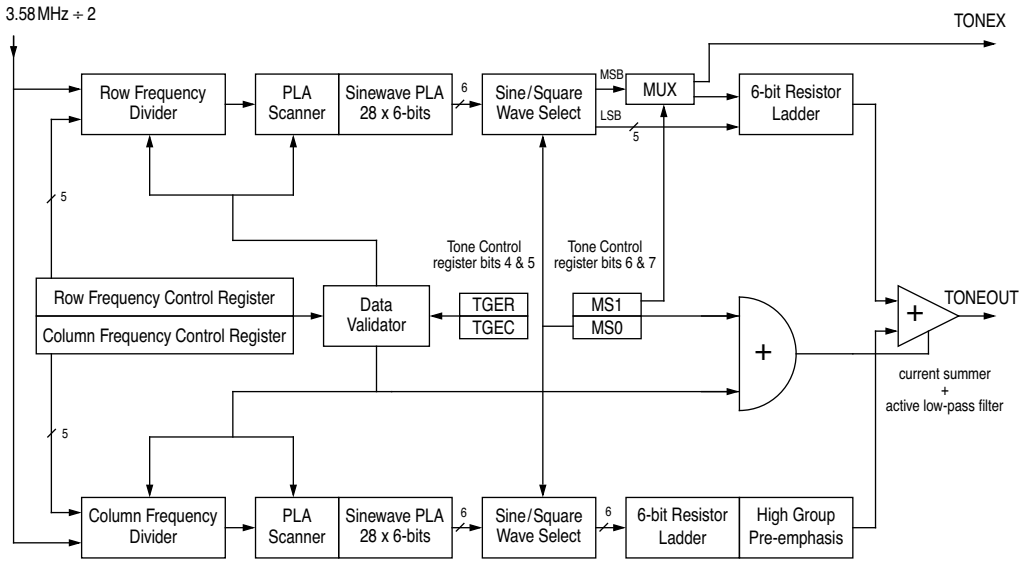


Figure 9-1 DTMF/Melody Generator Block Diagram

sampling rate of the desired “staircase sine wave”. This pulse train then clocks a divide-by-28 binary counter (PLA scanner) whose 28 decoded outputs sequentially scan 28 memory locations of a 28x6 sine wave generator (PLA) in 28 time steps (M). The 6 resulting digital sine wave bits are then fed separately to a 6-bit resistor ladder to produce a current signal.

The method for generating a square wave tone in a path is similar to that of a sine wave tone except that only the most significant bit of a sine wave PLA is fed to the 6-bit resistor ladder (the other 5 bits are masked by the Sine/Square wave select) to produce a current signal. The resulting square wave tone has exactly the same frequency and phase as a sine wave tone for the same frequency control register value.

After obtaining the current signals from the row and column paths, the row current signal is first attenuated by 2dB, and is then summed with the column current signal, and is finally fed to an active 7KHz low pass filter to reduce harmonic distortion. The resulting DTMF or melody signal is output to the TONEOUT pin, which is normally buffered to drive a buzzer.

The generator provides not only DTMF and melody but also a square wave pacifier tone (Tone). This signal is also extracted from the most significant bit of the sine wave PLA of the row path, but is not subjected to the filter. The ToneX signal is output to the TONEX pin, which is normally connected to a loudspeaker.

9.3 DMG Registers

The DMG has three registers, Row Frequency Control register and Column Frequency Control register, for row and column frequencies selection respectively; and Tone Control register for tone output control and mode selection.

9.3.1 Row Frequency Control Register (FCR)
Column Frequency Control Register (FCC)

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Row Frequency Control	$13				FCR4	FCR3	FCR2	FCR1	FCR0	000u uuuu

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Column Frequency Control	$14				FCC4	FCC3	FCC2	FCC1	FCC0	000u uuuu

FCR0-4 and FCC0-4 control the frequencies of the tone signals on the row and the column paths respectively. The bit description for DTMF and Melody tone generation are shown in Table 9-1 and Table 9-2 respectively.

Table 9-1 Bit Description for DTMF Generation

FCR	FCC	TONE	Standard Frequency (Hz)	Tone Output Frequency (Hz)	Frequency Deviation (%)
$00	See Note	f_{R1}	697	694.8	0.32
$01		f_{R2}	770	770.1	–0.02
$02		f_{R3}	852	854.2	–0.03
$03		f_{R4}	941	940.0	0.11
See note	$10	f_{C1}	1209	1206.0	0.244
	$11	f_{C2}	1336	1331.7	0.324
	$12	f_{C3}	1477	1486.5	–0.645
	$13	f_{C4}	1633	1639.0	–0.367

Note: The legal values in the FCR are illegal to the FCC, and vice versa. An illegal value to these registers will produce a tri-state at the TONEOUT output pin, and a logic high at the TONEX output pin.

Table 9-2 Bit Description for Melody Generation

FCR/FCC	Tone	Standard Frequency (Hz)	Tone Output Frequency (Hz)	Frequency Deviation (%)
$04	D#5	622.3	620.6	0.28
$05	E5	659.3	659.0	0.05
$06	F5	698.5	694.8	0.53
$07	F#5	740.0	743.3	–0.44
$08	G5	784.0	779.5	0.57
$09	G#5	830.6	830.1	0.06
$0A	A5	880.6	875.6	0.50
$0B	A#5	932.3	926.4	0.64
$0C	B5	987.8	983.4	0.45
$0D	C6	1046.5	1047.9	–0.13
$0E	C#6	1108.7	1102.1	0.60
$0F	D6	1174.7	1183.7	–0.77
$14	D#6	1224.5	1253.3	–0.71
$15	E6	1318.5	1331.7	–1.00
$16	F6	1396.9	1389.6	0.52
$17	F#6	1480.0	1486.5	–0.44
$18	G6	1568.0	1559.0	0.57
$19	G#6	1661.2	1682.1	–1.26
$1A	A6	1760.0	1775.6	–0.89
$1B	A#6	1864.7	1880.0	–0.82
$1C	B6	1975.5	1997.5	–1.11
$1D	C7	2093.0	2062.0	1.49
$1E	C#7	2217.5	2204.2	0.60
$1F	D7	2349.3	2367.4	–0.771

9.3.2 Tone Control Register (TNCR)

Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
$15	MS1	MS0	TGER	TGEC					0000 0000

This register controls the internal configuration and tone output timing of the DTMF/Melody Generator.

MS1, MS0 - Mode Select

These bits control the operating mode of the DTMF/Melody Generator. These are sine wave, square wave, ToneX, and square wave+ToneX modes. Table 9-3 shows the bit configurations.

When one mode is selected, the pin associated with that mode will be activated, but the other pin will remain at its idle state. The idle state for TONEOUT output pin is a tri-state, and TONEX output pin is a logic high. The final state of an active pin is dependent on the values of TGER, TGEC (see Table 9-4), FCR and FCC bits (when illegal values is input).

When both MS1 and MS0 are set, the generator can generate both single tone melody at the column path and ToneX at the row path simultaneously.

Table 9-3 DMG Operating Modes

MS1	MS0	Mode	TONEOUT output	TONEX output
0	0	Sine Wave	DTMF/Melody	High
0	1	Square Wave	Melody	High
1	0	ToneX	tri-state	ToneX
1	1	Square Wave+ToneX	Monotonic Melody	ToneX

TGER, TGEC - Tone Generation Enable for Row and Column Paths

When both bits are held low, the DMG is disabled by forcing the two frequency counters and the two PLA scanning counters to their reset states. The TONEOUT output is set to tri-state, the TONEX output is set to logic low, and the active filter is turned off by shutting down all related current sources to prevent DC power dissipation.

When a TGE bit for a path is held high (provided that the value in the frequency control register for that path is legal, and the mode chosen is not ToneX mode) the generator is enabled. All the counters associated with that path are then run from their reset states, and the active filter is turned on to allow generated tone of that path to be output.

In DTMF dialling, the row and column tone values are first entered to the FCR and FCC registers, and then the TGER and TGEC bits are set or reset simultaneously to achieve dual tone multiple frequency. Similarly, in melody generation, one path is chosen as the high part, and the other, the low part. The TGER and TGEC bits are then set and reset according to the rhythm required by the musical tune. Of course, one can exhibits only single tone melody by disabling either TGER or TGEC permanently. The DTMF column and row frequency tones can also be output separately for testing by enabling just the one path.

Table 9-4 Effect of Tone Generation Enable on DMG

TGER	TGEC	Row Path	Column Path	Filter	Tone
0	0	Off	Off	Off	Silent
0	1	Off	Active	Active	Single
1	0	Active	Off	Active*	Single
1	1	Active	Active	Active	Dual*

* In ToneX mode, the filter is off and only single tone can be generated.

Note: The reset state of a frequency counter defines the time=0 state of a time step, whereas the PLA scanning counters at its reset state scanning the memory location contained the DC value of staircase sine wave.

9.4 Programming the DMG

The recommended operating procedures for the DMG are described in the below paragraphs.

Since the TONEOUT pin is an open-collector output, an external pull-up resistor of 1K to 10KΩ is required (see Section 13 - Electrical Characteristics).

9.4.1 DTMF Dialling

To operate DTMF dialling, the sine wave mode selected. The required dual-tone (digit) are selected through the FCR and FCC registers, and are thus output to TONEOUT pin by setting both TGER and TGEC bits simultaneously for a period of 80ms. After generating a dual-tone, an inter-digit delay, which is produced by tri-stating the TONEOUT output, of another 80ms before the next dual-tone (digit) is output. This can be achieved by clearing both the TGER and TGEC bits simultaneously, or by writing an illegal value to FCR or FCC registers.

9.4.2 Melody Generation

For melody generation, either sine wave or square wave mode can be selected for full programmability. The sine wave has a flute like sound, while the square wave possesses much richer harmonics. The required tones are selected through the FCR and FCC registers. The selected tone is generated when the corresponding TGER or TGEC bit is set.

9.4.3 ToneX Generation

To operate ToneX generation, the ToneX mode is selected. The required tone is selected through the FCR register. The timing of the tone can be controlled by the TGER bit.

9.4.4 Melody+ToneX Generation

To operate Melody+ToneX generation, the ToneX+Melody mode is selected. The frequencies of the Melody tone and ToneX are selected through the FCR and FCC registers respectively, whereas the timings of the Melody and ToneX are controlled separately by TGER and TGEC respectively.

10
CPU CORE AND INSTRUCTION SET

This section provides a description of the CPU core registers, the instruction set and the addressing modes of the MC68HC05F8.

10.1 Registers

The MCU contains five registers, as shown in the programming model of Figure 10-1. The interrupt stacking order is shown in Figure 10-2.

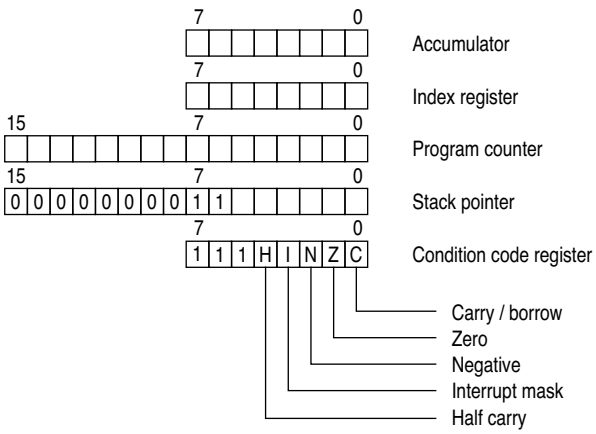


Figure 10-1 Programming model

10.1.1 Accumulator (A)

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

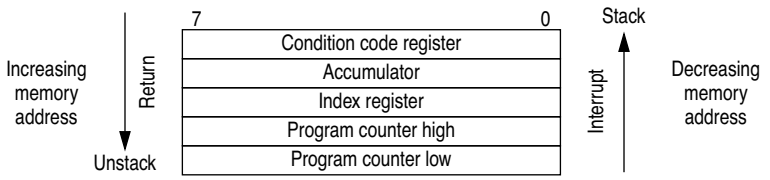


Figure 10-2 Stacking order

10.1.2 Index register (X)

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

10.1.3 Program counter (PC)

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

10.1.4 Stack pointer (SP)

The stack pointer is a 16-bit register, which 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 is then decremented as data is pushed onto the stack and incremented as data is pulled from the stack.

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

10.1.5 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 individually tested by a program, and specific actions can be taken as a result of their state. Each bit is explained in the following paragraphs.

Half carry (H)

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

Interrupt (I)

When this bit is set, all maskable interrupts are masked. If an interrupt occurs while this bit is set, the interrupt is latched and remains pending until the interrupt bit is cleared.

Negative (N)

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

Zero (Z)

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

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 is also affected during bit test and branch instructions and during shifts and rotates.

10.2 Instruction set

The MCU has a set of 62 basic instructions. They can be grouped into five different types as follows:

- Register/memory
- Read/modify/write
- Branch
- Bit manipulation
- Control

The following paragraphs briefly explain each type. All the instructions within a given type are presented in individual tables.

This MCU uses all the instructions available in the M146805 CMOS family plus one more: the unsigned multiply (MUL) instruction. This instruction allows unsigned multiplication of the contents of the accumulator (A) and the index register (X). The high-order product is then stored in the index register and the low-order product is stored in the accumulator. A detailed definition of the MUL instruction is shown in Table 10-1.

10.2.1 Register/memory Instructions

Most of these instructions use two operands. The first operand is either the accumulator or the index register. The second 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 Table 10-2 for a complete list of register/memory instructions.

10.2.2 Branch instructions

These instructions cause the program to branch if a particular condition is met; otherwise, no operation is performed. Branch instructions are two-byte instructions. Refer to Table 10-3.

10.2.3 Bit manipulation instructions

The MCU can set or clear any writable bit that resides in the first 256 bytes of the memory space (page 0). All port data and data direction registers, timer and serial interface registers, control/status registers and a portion of the on-chip RAM reside in page 0. An additional feature allows the software to test and branch on the state of any bit within these locations. The bit set, bit clear, bit test and branch functions are all implemented with single instructions. For the test and branch instructions, the value of the bit tested is also placed in the carry bit of the condition code register. Refer to Table 10-4.

10.2.4 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 this sequence of reading, modifying and writing, since it does not modify the value. Refer to Table 10-5 for a complete list of read/modify/write instructions.

10.2.5 Control instructions

These instructions are register reference instructions and are used to control processor operation during program execution. Refer to Table 10-6 for a complete list of control instructions.

10.2.6 Tables

Tables for all the instruction types listed above follow. In addition there is a complete alphabetical listing of all the instructions (see Table 10-7), and an opcode map for the instruction set of the M68HC05 MCU family (see Table 10-8).

Table 10-1 MUL instruction

Operation	**X:A ← X*A**			
Description	Multiplies the eight bits in the index register by the eight bits in the accumulator and places the 16-bit result in the concatenated accumulator and index register.			
Condition codes	H : Cleared I : Not affected N : Not affected Z : Not affected C : Cleared			
Source	MUL			
Form	Addressing mode	Cycles	Bytes	Opcode
	Inherent	11	1	$42

Table 10-2 Register/memory instructions

		Addressing modes																	
		Immediate			**Direct**			**Extended**			**Indexed (no offset)**			**Indexed (8-bit offset)**			**Indexed (16-bit offset)**		
Function	**Mnemonic**	**Opcode**	**# Bytes**	**# Cycles**	**Opcode**	**# Bytes**	**# Cycles**	**Opcode**	**# Bytes**	**# Cycles**	**Opcode**	**# Bytes**	**# Cycles**	**Opcode**	**# Bytes**	**# Cycles**	**Opcode**	**# Bytes**	**# Cycles**
Load A from memory	LDA	A6	2	2	B6	2	3	C6	3	4	F6	1	3	E6	2	4	D6	3	5
Load X from memory	LDX	AE	2	2	BE	2	3	CE	3	4	FE	1	3	EE	2	4	DE	3	5
Store A in memory	STA				B7	2	4	C7	3	5	F7	1	4	E7	2	5	D7	3	6
Store X in memory	STX				BF	2	4	CF	3	5	FF	1	4	EF	2	5	DF	3	6
Add memory to A	ADD	AB	2	2	BB	2	3	CB	3	4	FB	1	3	EB	2	4	DB	3	5
Add memory and carry to A	ADC	A9	2	2	B9	2	3	C9	3	4	F9	1	3	E9	2	4	D9	3	5
Subtract memory	SUB	A0	2	2	B0	2	3	C0	3	4	F0	1	3	E0	2	4	D0	3	5
Subtract memory from A with borrow	SBC	A2	2	2	B2	2	3	C2	3	4	F2	1	3	E2	2	4	D2	3	5
AND memory with A	AND	A4	2	2	B4	2	3	C4	3	4	F4	1	3	E4	2	4	D4	3	5
OR memory with A	ORA	AA	2	2	BA	2	3	CA	3	4	FA	1	3	EA	2	4	DA	3	5
Exclusive OR memory with A	EOR	A8	2	2	B8	2	3	C8	3	4	F8	1	3	E8	2	4	D8	3	5
Arithmetic compare A with memory	CMP	A1	2	2	B1	2	3	C1	3	4	F1	1	3	E1	2	4	D1	3	5
Arithmetic compare X with memory	CPX	A3	2	2	B3	2	3	C3	3	4	F3	1	3	E3	2	4	D3	3	5
Bit test memory with A (logical compare)	BIT	A5	2	2	B5	2	3	C5	3	4	F5	1	3	E5	2	4	D5	3	5
Jump unconditional	JMP				BC	2	2	CC	3	3	FC	1	2	EC	2	3	DC	3	4
Jump to subroutine	JSR				BD	2	5	CD	3	6	FD	1	5	ED	2	6	DD	3	7

Table 10-3 Branch instructions

		Relative addressing mode		
Function	**Mnemonic**	**Opcode**	**# Bytes**	**# Cycles**
Branch always	BRA	20	2	3
Branch never	BRN	21	2	3
Branch if higher	BHI	22	2	3
Branch if lower or same	BLS	23	2	3
Branch if carry clear	BCC	24	2	3
(Branch if higher or same)	(BHS)	24	2	3
Branch if carry set	BCS	25	2	3
(Branch if lower)	(BLO)	25	2	3
Branch if not equal	BNE	26	2	3
Branch if equal	BEQ	27	2	3
Branch if half carry clear	BHCC	28	2	3
Branch if half carry set	BHCS	29	2	3
Branch if plus	BPL	2A	2	3
Branch if minus	BMI	2B	2	3
Branch if interrupt mask bit is clear	BMC	2C	2	3
Branch if interrupt mask bit is set	BMS	2D	2	3
Branch if interrupt line is low	BIL	2E	2	3
Branch if interrupt line is high	BIH	2F	2	3
Branch to subroutine	BSR	AD	2	6

Table 10-4 Bit manipulation instructions

		Addressing modes					
		Bit set/clear			Bit test and branch		
Function	**Mnemonic**	**Opcode**	**# Bytes**	**# Cycles**	**Opcode**	**# Bytes**	**# Cycles**
Branch if bit n is set	BRSET n (n=0–7)				2•n	3	5
Branch if bit n is clear	BRCLR n (n=0–7)				01+2•n	3	5
Set bit n	BSET n (n=0–7)	10+2•n	2	5			
Clear bit n	BCLR n (n=0–7)	11+2•n	2	5			

Table 10-5 Read/modify/write instructions

		Addressing modes														
		Inherent (A)			Inherent (X)			Direct			Indexed (no offset)			Indexed (8-bit offset)		
Function	Mnemonic	Opcode	# Bytes	# Cycles	Opcode	# Bytes	# Cycles	Opcode	# Bytes	# Cycles	Opcode	# Bytes	# Cycles	Opcode	# Bytes	# Cycles
Increment	INC	4C	1	3	5C	1	3	3C	2	5	7C	1	5	6C	2	6
Decrement	DEC	4A	1	3	5A	1	3	3A	2	5	7A	1	5	6A	2	6
Clear	CLR	4F	1	3	5F	1	3	3F	2	5	7F	1	5	6F	2	6
Complement	COM	43	1	3	53	1	3	33	2	5	73	1	5	63	2	6
Negate (two's complement)	NEG	40	1	3	50	1	3	30	2	5	70	1	5	60	2	6
Rotate left through carry	ROL	49	1	3	59	1	3	39	2	5	79	1	5	69	2	6
Rotate right through carry	ROR	46	1	3	56	1	3	36	2	5	76	1	5	66	2	6
Logical shift left	LSL	48	1	3	58	1	3	38	2	5	78	1	5	68	2	6
Logical shift right	LSR	44	1	3	54	1	3	34	2	5	74	1	5	64	2	6
Arithmetic shift right	ASR	47	1	3	57	1	3	37	2	5	77	1	5	67	2	6
Test for negative or zero	TST	4D	1	3	5D	1	3	3D	2	4	7D	1	4	6D	2	5
Multiply	MUL	42	1	11												

Table 10-6 Control instructions

		Inherent addressing mode		
Function	Mnemonic	Opcode	# Bytes	# Cycles
Transfer A to X	TAX	97	1	2
Transfer X to A	TXA	9F	1	2
Set carry bit	SEC	99	1	2
Clear carry bit	CLC	98	1	2
Set interrupt mask bit	SEI	9B	1	2
Clear interrupt mask bit	CLI	9A	1	2
Software interrupt	SWI	83	1	10
Return from subroutine	RTS	81	1	6
Return from interrupt	RTI	80	1	9
Reset stack pointer	RSP	9C	1	2
No-operation	NOP	9D	1	2
Stop	STOP	8E	1	2
Wait	WAIT	8F	1	2

Table 10-7 Instruction set

Mnemonic	Addressing modes										Condition codes				
	INH	IMM	DIR	EXT	REL	IX	IX1	IX2	BSC	BTB	H	I	N	Z	C
ADC											◊	•	◊	◊	◊
ADD											◊	•	◊	◊	◊
AND											•	•	◊	◊	•
ASL											•	•	◊	◊	◊
ASR											•	•	◊	◊	◊
BCC											•	•	•	•	•
BCLR											•	•	•	•	•
BCS											•	•	•	•	•
BEQ											•	•	•	•	•
BHCC											•	•	•	•	•
BHCS											•	•	•	•	•
BHI											•	•	•	•	•
BHS											•	•	•	•	•
BIH											•	•	•	•	•
BIL											•	•	•	•	•
BIT											•	•	◊	◊	•
BLO											•	•	•	•	•
BLS											•	•	•	•	•
BMC											•	•	•	•	•
BMI											•	•	•	•	•
BMS											•	•	•	•	•
BNE											•	•	•	•	•
BPL											•	•	•	•	•
BRA											•	•	•	•	•
BRN											•	•	•	•	•
BRCLR											•	•	•	•	◊
BRSET											•	•	•	•	◊
BSET											•	•	•	•	•
BSR											•	•	•	•	•
CLC											•	•	•	•	0
CLI											•	0	•	•	•
CLR											•	•	0	1	•
CMP											•	•	◊	◊	◊

Address mode abbreviations

BSC	Bit set/clear	IMM	Immediate
BTB	Bit test & branch	IX	Indexed (no offset)
DIR	Direct	IX1	Indexed, 1 byte offset
EXT	Extended	IX2	Indexed, 2 byte offset
INH	Inherent	REL	Relative

Not implemented

Condition code symbols

H	Half carry (from bit 3)	◊	Tested and set if true, cleared otherwise
I	Interrupt mask	•	Not affected
N	Negate (sign bit)	?	Load CCR from stack
Z	Zero	0	Cleared
C	Carry/borrow	1	Set

Table 10-7 Instruction set (Continued)

Mnemonic	Addressing modes										Condition codes				
	INH	IMM	DIR	EXT	REL	IX	IX1	IX2	BSC	BTB	H	I	N	Z	C
COM											•	•	◊	◊	1
CPX											•	•	◊	◊	◊
DEC											•	•	◊	◊	•
EOR											•	•	◊	◊	•
INC											•	•	◊	◊	•
JMP											•	•	•	•	•
JSR											•	•	•	•	•
LDA											•	•	◊	◊	•
LDX											•	•	◊	◊	•
LSL											•	•	◊	◊	◊
LSR											•	•	0	◊	◊
MUL											0	•	•	•	0
NEG											•	•	◊	◊	◊
NOP											•	•	•	•	•
ORA											•	•	◊	◊	•
ROL											•	•	◊	◊	◊
ROR											•	•	◊	◊	◊
RSP											•	•	•	•	•
RTI											?	?	?	?	?
RTS											•	•	•	•	•
SBC											•	•	◊	◊	◊
SEC											•	•	•	•	1
SEI											•	1	•	•	•
STA											•	•	◊	◊	•
STOP											•	0	•	•	•
STX											•	•	◊	◊	•
SUB											•	•	◊	◊	◊
SWI											•	1	•	•	•
TAX											•	•	•	•	•
TST											•	•	◊	◊	•
TXA											•	•	•	•	•
WAIT											•	0	•	•	•

Address mode abbreviations

BSC	Bit set/clear	IMM	Immediate
BTB	Bit test & branch	IX	Indexed (no offset)
DIR	Direct	IX1	Indexed, 1 byte offset
EXT	Extended	IX2	Indexed, 2 byte offset
INH	Inherent	REL	Relative

Not implemented

Condition code symbols

H	Half carry (from bit 3)	◊	Tested and set if true, cleared otherwise
I	Interrupt mask	•	Not affected
N	Negate (sign bit)	?	Load CCR from stack
Z	Zero	0	Cleared
C	Carry/borrow	1	Set

Table 10-8 M68HC05 opcode map

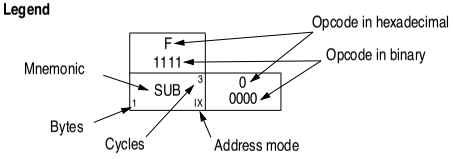
	Bit manipulation		Branch	Read/modify/write					Control		Register/memory						
	BTB	BSC	REL	DIR	INH	INH	IX1	IX	INH	INH	IMM	DIR	EXT	IX2	IX1	IX	
High / Low	0 0000	1 0001	2 0010	3 0011	4 0100	5 0101	6 0110	7 0111	8 1000	9 1001	A 1010	B 1011	C 1100	D 1101	E 1110	F 1111	High / Low
0 0000	BRSET0 5 3 BTB	BSET0 5 2 BSC	BRA 3 2 REL	NEG 5 2 DIR	NEGA 3 1 INH	NEGX 3 1 INH	NEG 6 2 IX1	NEG 5 1 IX	RTI 9 1 INH		SUB 2 2 IMM	SUB 3 2 DIR	SUB 4 3 EXT	SUB 5 3 IX2	SUB 4 2 IX1	SUB 3 1 IX	0 0000
1 0001	BRCLR0 5 3 BTB	BCLR0 5 2 BSC	BRN 3 2 REL						RTS 6 1 INH		CMP 2 2 IMM	CMP 3 2 DIR	CMP 4 3 EXT	CMP 5 3 IX2	CMP 4 2 IX1	CMP 3 1 IX	1 0001
2 0010	BRSET1 5 3 BTB	BSET1 5 2 BSC	BHI 3 2 REL		MUL 11 1 INH						SBC 2 2 IMM	SBC 3 2 DIR	SBC 4 3 EXT	SBC 5 3 IX2	SBC 4 2 IX1	SBC 3 1 IX	2 0010
3 0011	BRCLR1 5 3 BTB	BCLR1 5 2 BSC	BLS 3 2 REL	COM 5 2 DIR	COMA 3 1 INH	COMX 3 1 INH	COM 6 2 IX1	COM 5 1 IX	SWI 10 1 INH		CPX 2 2 IMM	CPX 3 2 DIR	CPX 4 3 EXT	CPX 5 3 IX2	CPX 4 2 IX1	CPX 3 1 IX	3 0011
4 0100	BRSET2 5 3 BTB	BSET2 5 2 BSC	BCC 3 2 REL	LSR 5 2 DIR	LSRA 3 1 INH	LSRX 3 1 INH	LSR 6 2 IX1	LSR 5 1 IX			AND 2 2 IMM	AND 3 2 DIR	AND 4 3 EXT	AND 5 3 IX2	AND 4 2 IX1	AND 3 1 IX	4 0100
5 0101	BRCLR2 5 3 BTB	BCLR2 5 2 BSC	BCS 3 2 REL								BIT 2 2 IMM	BIT 3 2 DIR	BIT 4 3 EXT	BIT 5 3 IX2	BIT 4 2 IX1	BIT 3 1 IX	5 0101
6 0110	BRSET3 5 3 BTB	BSET3 5 2 BSC	BNE 3 2 REL	ROR 5 2 DIR	RORA 3 1 INH	RORX 3 1 INH	ROR 6 2 IX1	ROR 5 1 IX			LDA 2 2 IMM	LDA 3 2 DIR	LDA 4 3 EXT	LDA 5 3 IX2	LDA 4 2 IX1	LDA 3 1 IX	6 0110
7 0111	BRCLR3 5 3 BTB	BCLR3 5 2 BSC	BEQ 3 2 REL	ASR 5 2 DIR	ASRA 3 1 INH	ASRX 3 1 INH	ASR 6 2 IX1	ASR 5 1 IX		TAX 2 1 INH		STA 4 2 DIR	STA 5 3 EXT	STA 6 3 IX2	STA 5 2 IX1	STA 4 1 IX	7 0111
8 1000	BRSET4 5 3 BTB	BSET4 5 2 BSC	BHCC 3 2 REL	LSL 5 2 DIR	LSLA 3 1 INH	LSLX 3 1 INH	LSL 6 2 IX1	LSL 5 1 IX		CLC 2 1 INH	EOR 2 2 IMM	EOR 3 2 DIR	EOR 4 3 EXT	EOR 5 3 IX2	EOR 4 2 IX1	EOR 3 1 IX	8 1000
9 1001	BRCLR4 5 3 BTB	BCLR4 5 2 BSC	BHCS 3 2 REL	ROL 5 2 DIR	ROLA 3 1 INH	ROLX 3 1 INH	ROL 6 2 IX1	ROL 5 1 IX		SEC 2 1 INH	ADC 2 2 IMM	ADC 3 2 DIR	ADC 4 3 EXT	ADC 5 3 IX2	ADC 4 2 IX1	ADC 3 1 IX	9 1001
A 1010	BRSET5 5 3 BTB	BSET5 5 2 BSC	BPL 3 2 REL	DEC 5 2 DIR	DECA 3 1 INH	DECX 3 1 INH	DEC 6 2 IX1	DEC 5 1 IX		CLI 2 1 INH	ORA 2 2 IMM	ORA 3 2 DIR	ORA 4 3 EXT	ORA 5 3 IX2	ORA 4 2 IX1	ORA 3 1 IX	A 1010
B 1011	BRCLR5 5 3 BTB	BCLR5 5 2 BSC	BMI 3 2 REL							SEI 2 1 INH	ADD 2 2 IMM	ADD 3 2 DIR	ADD 4 3 EXT	ADD 5 3 IX2	ADD 4 2 IX1	ADD 3 1 IX	B 1011
C 1100	BRSET6 5 3 BTB	BSET6 5 2 BSC	BMC 3 2 REL	INC 5 2 DIR	INCA 3 1 INH	INCX 3 1 INH	INC 6 2 IX1	INC 5 1 IX		RSP 2 1 INH		JMP 2 2 DIR	JMP 3 3 EXT	JMP 4 3 IX2	JMP 3 2 IX1	JMP 2 1 IX	C 1100
D 1101	BRCLR6 5 3 BTB	BCLR6 5 2 BSC	BMS 3 2 REL	TST 4 2 DIR	TSTA 3 1 INH	TSTX 3 1 INH	TST 5 2 IX1	TST 4 1 IX		NOP 2 1 INH	BSR 6 2 REL	JSR 5 2 DIR	JSR 6 3 EXT	JSR 7 3 IX2	JSR 6 2 IX1	JSR 5 1 IX	D 1101
E 1110	BRSET7 5 3 BTB	BSET7 5 2 BSC	BIL 3 2 REL						STOP 2 1 INH		LDX 2 2 IMM	LDX 3 2 DIR	LDX 4 3 EXT	LDX 5 3 IX2	LDX 4 2 IX1	LDX 3 1 IX	E 1110
F 1111	BRCLR7 5 3 BTB	BCLR7 5 2 BSC	BIH 3 2 REL	CLR 5 2 DIR	CLRA 3 1 INH	CLRX 3 1 INH	CLR 6 2 IX1	CLR 5 1 IX	WAIT 2 1 INH	TXA 2 1 INH		STX 4 2 DIR	STX 5 3 EXT	STX 6 3 IX2	STX 5 2 IX1	STX 4 1 IX	F 1111

Abbreviations for address modes and registers

BSC	Bit set/clear	IX	Indexed (no offset)
BTB	Bit test and branch	IX1	Indexed, 1 byte (8-bit) offset
DIR	Direct	IX2	Indexed, 2 byte (16-bit) offset
EXT	Extended	REL	Relative
INH	Inherent	A	Accumulator
IMM	Immediate	X	Index register

Not implemented

Legend



10.3 Addressing modes

Ten different addressing modes provide programmers with the flexibility to optimize their 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) enable access to tables throughout memory. Short absolute (direct) and long absolute (extended) addressing are also included. One or two byte direct addressing instructions access all data bytes in most applications. Extended addressing permits jump instructions to reach all memory locations.

The term ‘effective address’ (EA) is used in describing the various addressing modes. The effective address is defined as the address from which the argument for an instruction is fetched or stored. The ten addressing modes of the processor are described below. Parentheses are used to indicate ‘contents of’ the location or register referred to. For example, (PC) indicates the contents of the location pointed to by the PC (program counter). An arrow indicates ‘is replaced by’ and a colon indicates concatenation of two bytes. For additional details and graphical illustrations, refer to the ***M68HC05 Applications Guide***.

10.3.1 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 or accumulator, as well as the control instruction, with no other arguments are included in this mode. These instructions are one byte long.

10.3.2 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 (e.g. a constant used to initialize a loop counter).

$$EA = PC+1;\ PC \leftarrow PC+2$$

10.3.3 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 with a single two-byte instruction.

$$EA = (PC+1);\ PC \leftarrow PC+2$$
$$\text{Address bus high} \leftarrow 0;\ \text{Address bus low} \leftarrow (PC+1)$$

10.3.4 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 three-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 short form of the instruction.

EA = (PC+1):(PC+2); PC ← PC+3
Address bus high ← (PC+1); Address bus low ← (PC+2)

10.3.5 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. 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.

EA = X; PC ← PC+1
Address bus high ← 0; Address bus low ← X

10.3.6 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. Therefore the operand can be located anywhere within the lowest 511 memory locations. This addressing mode is useful for selecting the mth element in an n element table.

EA = X+(PC+1); PC ← PC+2
Address bus high ← K; Address bus low ← X+(PC+1)
where K = the carry from the addition of X and (PC+1)

10.3.7 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 three-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.

EA = X+[(PC+1):(PC+2)]; PC ← PC+3
Address bus high ← (PC+1)+K; Address bus low ← X+(PC+2)
where K = the carry from the addition of X and (PC+2)

10.3.8 Relative

The relative addressing mode is only used in branch instructions. In relative addressing, the contents of the 8-bit signed byte (the offset) following the opcode are 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 –126 to +129 from the opcode address. 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.

EA = PC+2+(PC+1); PC ← EA if branch taken;
otherwise EA = PC ← PC+2

10.3.9 Bit set/clear

In the bit set/clear addressing mode, the bit to be set or cleared is part of the opcode. The byte following the opcode specifies the address 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 two-byte instruction.

EA = (PC+1); PC ← PC+2
Address bus high ← 0; Address bus low ← (PC+1)

10.3.10 Bit test and branch

The bit test and branch addressing mode is a combination of direct addressing and relative addressing. The bit 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 (EA1). The signed relative 8-bit offset in the third byte (EA2) is added to the PC if the specified bit is set or cleared in the specified memory location. This single three-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 branch is from –125 to +130 from the opcode address. The state of the tested bit is also transferred to the carry bit of the condition code register.

EA1 = (PC+1); PC ← PC+2
Address bus high ← 0; Address bus low ← (PC+1)
EA2 = PC+3+(PC+2); PC ← EA2 if branch taken;
otherwise PC ← PC+3

THIS PAGE LEFT BLANK INTENTIONALLY

11
LOW POWER MODES

The STOP and WAIT instructions have different effects on the Timers, Serial Peripheral Interface (SPI), and DTMF/Melody Generator (DMG). These are discussed in the following paragraphs.

11.1 Stop Mode

When the processor executes the STOP instruction, the internal clock is turned off. This halts all internal CPU processing, including the operation of the Programmable Timer, SPI and DMG. The I bit in the Condition Code register is cleared to enable external interrupts (INTE1, INTE2 and KEYE bits are unaltered). All registers and memory remain unaltered, and all input/output lines remain unchanged.

The MCU is exited from Stop mode by an interrupt on either $\overline{\text{IRQ1}}$ or $\overline{\text{IRQ2}}$, or any keyboard interrupts, or any resets (logic low on $\overline{\text{RESET}}$ pin or a power-on reset). On exit from Stop mode, the program counter is loaded with the corresponding interrupt vector (see Table 5-1). The effects of the Stop mode on each of the MCU peripheral systems are described separately.

11.1.1 Timer A during Stop Mode

When Stop mode is entered, the timer A (programmable timer) counter stops counting (the internal processor clock is stopped) and remains at that particular count value until the Stop mode is exited. If the exit was caused by reset, the counter is forced to $FFFC. If the Stop mode is exited by an interrupt ($\overline{\text{IRQ1}}$, $\overline{\text{IRQ2}}$, or keyboard interrupt), the counter resumes counting from the value when it entered the Stop mode. Another feature of the programmable timer in the Stop mode is, that if at least one valid input capture edge occurs at the TCAP pin, the input capture detect circuitry is armed. This action does not set any timer flags or “wake up” the MCU, but when the MCU does “wake up” there will be an active input capture flag (and data) from that first valid edge which occurred during the Stop mode. Notice that an exit by a reset will reset the entire MCU and thus, this function on the TCAP will not happen.

11.1.2 Timer B during Stop Mode

When Stop mode is entered, the timer B (reloadable timer) counter stops counting (the internal processor clock is stopped) and remains at that particular count value until the Stop mode is exited. If the exit was caused by reset, the reloadable timer is disabled. If the Stop mode is exited by an interrupt ($\overline{IRQ1}$, $\overline{IRQ2}$, or keyboard interrupt), the counter resumes counting from the value when it entered the Stop mode.

11.1.3 SPI during Stop Mode

When the Stop mode is entered, the baud rate generator driving the SPI shuts down. This stops all master mode SPI operations, thus the master SPI is unable to transmit or receive any data. If the STOP instruction is executed during an SPI transfer, that transfer is halted until the MCU exits the Stop mode by an interrupt ($\overline{IRQ1}$, $\overline{IRQ2}$, or keyboard interrupt). If the Stop mode is exited by a reset, the appropriate control/status bits are cleared and the SPI is disabled. If the device is in the slave mode when the STOP instruction is executed, the slave SPI will still operate. It can still accept data, clock information, and transmit data back to a master device, but no flags are set at the end of the transmission until the Stop mode is exited by an interrupt. The user should be careful when using the SPI as slave during the Stop mode because data protection features are not active (e.g. write collision).

It should also be noted that when the MCU is in the Stop mode, the enabled output drivers (TCMP, SDO, SDI, and SCK ports) remain active, and any sourcing currents from these outputs will be part of the total supply current required by the device.

11.1.4 DMG during Stop Mode

When the Stop mode is entered, all counters which generate the timings for the DTMF and Melody, and all current sources of the active filter will be shut down. The TONEOUT pin of the DMG will be tri-stated and the TONEX pin will be at logic high. All DMG operations are halted.

11.1.5 COP during Stop Mode

If the COP system is enabled and the “kill” watchdog timer feature is not activated, the watchdog timer will continue to run in Stop mode, and eventually time-out, causing a reset to the MCU.

If the COP system is enabled and the “kill” watchdog timer is activated, a STOP instruction will reset the watchdog timer and disable the COP system.

If the WDTE bit is set, when the MCU exits Stop mode (by an interrupt), the COP system is automatically enabled and the watchdog timer counter is loaded with the initial value. The COP system remains inactive if WDTE bit is cleared.

11.2 Wait Mode

When the MCU enters the Wait mode, the CPU clock is halted. All CPU activities and the DTMF/Melody Generation are halted, as in Stop mode; however, the timers (A and B) and SPI system remain active. An interrupt from the timer, keyboard, SPI, or $\overline{IRQ1}$/$\overline{IRQ2}$ causes the processor to exit the Wait mode. A reset will also take the MCU out of Wait mode.

The operation of the COP system in Wait mode is as for Stop mode.

The Wait mode power consumption depends on how many systems are active. The power consumption will be the least when the SPI and timer are disabled. If a non-reset exit from the Wait mode is performed (e.g. timer overflow interrupt exit), the state of the remaining systems will be unchanged. If a reset exit from the wait mode is performed, all the systems revert to the disabled reset state.

THIS PAGE LEFT BLANK INTENTIONALLY

12
OPERATING MODES

The MC68HC05F8/*MC68HC705F8* MCU has two modes of operation, the User Mode and the Self-Check/*Bootstrap* Mode. Figure 12-1 shows the flowchart of entry to these two modes, and Table 12-1 shows operating mode selection.

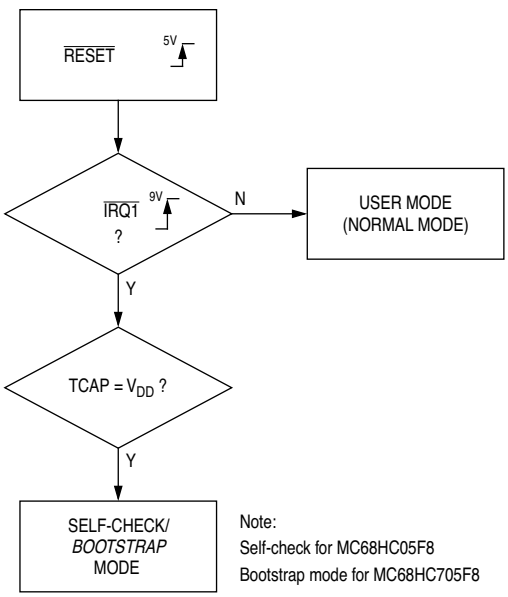


Figure 12-1 Flowchart of Mode Entering

Table 12-1 Mode Selection

$\overline{\text{RESET}}$	$\overline{\text{IRQ1}}$	TCAP	MODE
5V	V_{SS} to V_{DD}	V_{SS} to V_{DD}	USER
5V	9V +9V Rising Edge*	V_{DD}	SELF-CHECK/ *BOOTSTRAP*
* Minimum hold time should be 2 clock cycles, after that it can be used as a normal $\overline{\text{IRQ1}}$ function pin.			

12.1 User Mode (Normal Operation)

The normal operating mode of the MC68HC05F8/*MC68HC705F8* is the user mode. The user mode is entered if the $\overline{\text{RESET}}$ line is brought low, and the $\overline{\text{IRQ1}}$ pin is within its normal operating range (V_{SS} to V_{DD}), the rising edge of the $\overline{\text{RESET}}$ will cause the MCU to enter the user mode.

12.2 Self-Check Mode

The self-check mode is available on the MC68HC05F8 only, and is for the user to check device functions with an on-chip self-check program masked at location $FE00 to $FEDF under minimum hardware support. The self-check circuit is shown in Figure 12-3. Figure 12-2 is the criteria to enter self-check mode, where TCAP's condition is latched within first two clock cycles after the rising edge of the reset. TCAP can then be used for other purposes. After entering the self-check mode, CPU branches to the self-check program and carries out the self-check. Self-check is a repetitive test, i.e. if all parts are checked to be good, the CPU will repeat the self-check again. Therefore, the LEDs attached to Port A will be flashing if the device is good; else the combination of LEDs' on-off pattern will indicate which part of the device is suspected to be bad. Table 12-2 lists the LEDs' on-off patterns and their corresponding indications.

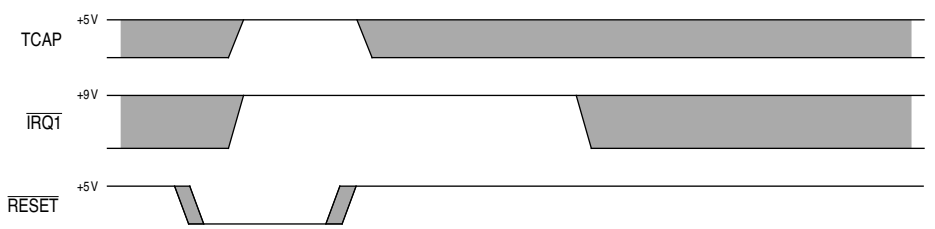


Figure 12-2 Self-Check Mode Timing

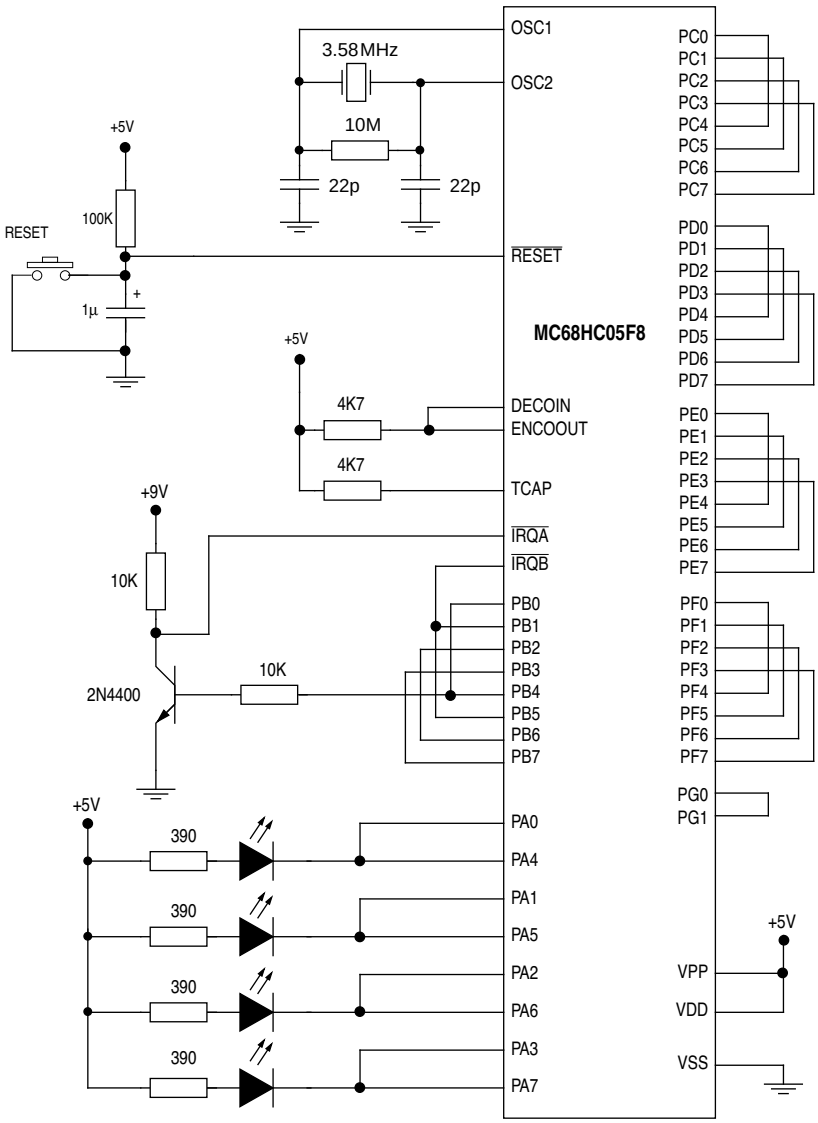


Figure 12-3 Self-Test Circuit

Table 12-2 Self-Check Report

PA3	PA2	PA1	PA0	REMARKS
1	1	1	1	Faulty part, port A bad
1	1	1	0	Bad I/O
1	1	0	1	Bad RAM
1	1	0	0	Bad ROM
1	0	1	1	Bad Timer A
1	0	1	0	Bad Timer B
1	0	0	1	Bad SPI
1	0	0	0	Bad MANCD
0	1	1	1	Bad Interrupts
Flashing				Good Device
All Others				Bad device, port A, etc.

1=LED off; 0=LED on.

12.3 Bootstrap Mode

The bootstrap mode is available on the MC68HC705F8 only, and it is a mean of self-programming its EPROM with minimal circuitry. It is entered on the rising edge of $\overline{\text{RESET}}$ if $\overline{\text{IRQ1}}$ pin is at $1.8V_{DD}$ and TCAP is at logic one. $\overline{\text{RESET}}$ must be held low for 4064 cycles after POR (power-on reset) or for a time t_{RL} for any other reset. Table 12-3 shows the options that are available once bootstrap mode is entered. The execution result is indicated by two LEDs. The EPROM programming circuit for bootstrap mode is shown in Figure 12-5.

Table 12-3 Bootstrap Mode Options

PB0	REMARKS
0	Program & Verify
1	Verify

12

12.3.1 EPROM Program Control Register

Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
$3F	0	0	0	0	0	0	LAT	EPGM	0000 0000

EPROM programming is controlled by the Program Control Register at location $3F.

LAT - Latch EPROM Data and Address

1 (set) – EPROM address and data buses configured for programming.

0 (clear) – EPROM address and data buses configured for normal reads.

LAT causes address and data buses to be latched when a write to EPROM is carried out. The EPROM cannot be read if LAT=1. This bit should not be set unless a programming voltage is applied to the VPP pin.

EPGM - EPROM Programming Mode Enable

1 (set) – Programming power connected to the EPROM array.

0 (clear) – Programming power disconnected from the EPROM array.

LAT and EPMG cannot be set on the same write operation. EPMG can only be set if LAT is set. EPMG is automatically cleared when LAT is cleared.

12.3.2 EPROM Programming Sequence

In the bootstrap, the user program contained in an external EPROM is copied into the internal EPROM of the MC68HC705F8 device (see Figure 12-5). The MC68HC705F8 device is inserted into the programming circuit as shown in Figure 12-5. Programming routine is selected via mode switch S1, and +5V and VPP power is applied to the programming circuitry. The MCU is removed from the reset state and placed in the run mode of operation via switch S4, and MCU control is transferred to the bootstrap ROM. The selected programming routine is then executed.

Programming sequence of events are as follows:

1) Place switch S4 to RESET position (switch close).
2) Select programming routine via switches S1.
3) Apply +5 V and VPP power to programming circuitry.
4) Place switch S4 to RUN position (switch open).
5) Programming routine is executed.
6) Place switch S4 to RESET position.
7) Remove VPP and +5 V power, or select and run new routine.

Once the bootstrap mode is entered, mode switch setting is scanned to establish the routine to be executed. The routines are:

– Program and Verify EPROM
– Verify EPROM Contents

Note: When programming the window part MC68HC705F8, the window should be covered up to prevent erratic device behaviour.

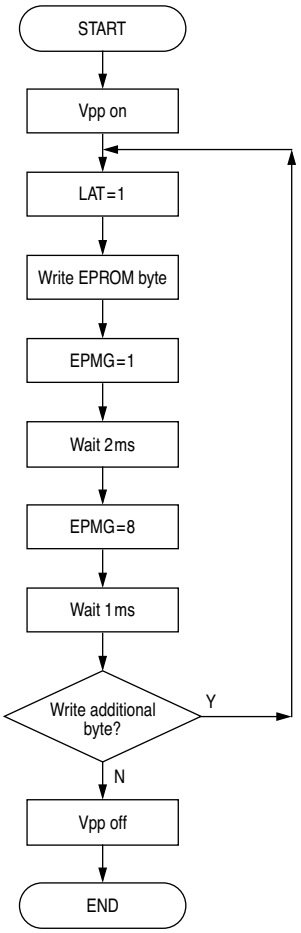


Figure 12-4 EPROM Programming Sequence

12.3.3 Program and Verify EPROM

In the Program and Verify EPROM routine, the contents of the external EPROM are copied into the EPROM areas of the MC68HC705F8 device. There is a direct correspondence of addresses between the two devices. Non-EPROM addresses are ignored so data contained in those areas are not accessed. Unprogrammed EPROM address locations should contain $FF to speed up the programming operation. During the programming routine the Program/Verify LED is illuminated. At the end of the programming routine, the LED is turned off, and the verification routine is entered. If the contents of the EPROM and external EPROM exactly match, then the Verified LED is illuminated. The verification routine stops if a discrepancy has been detected.

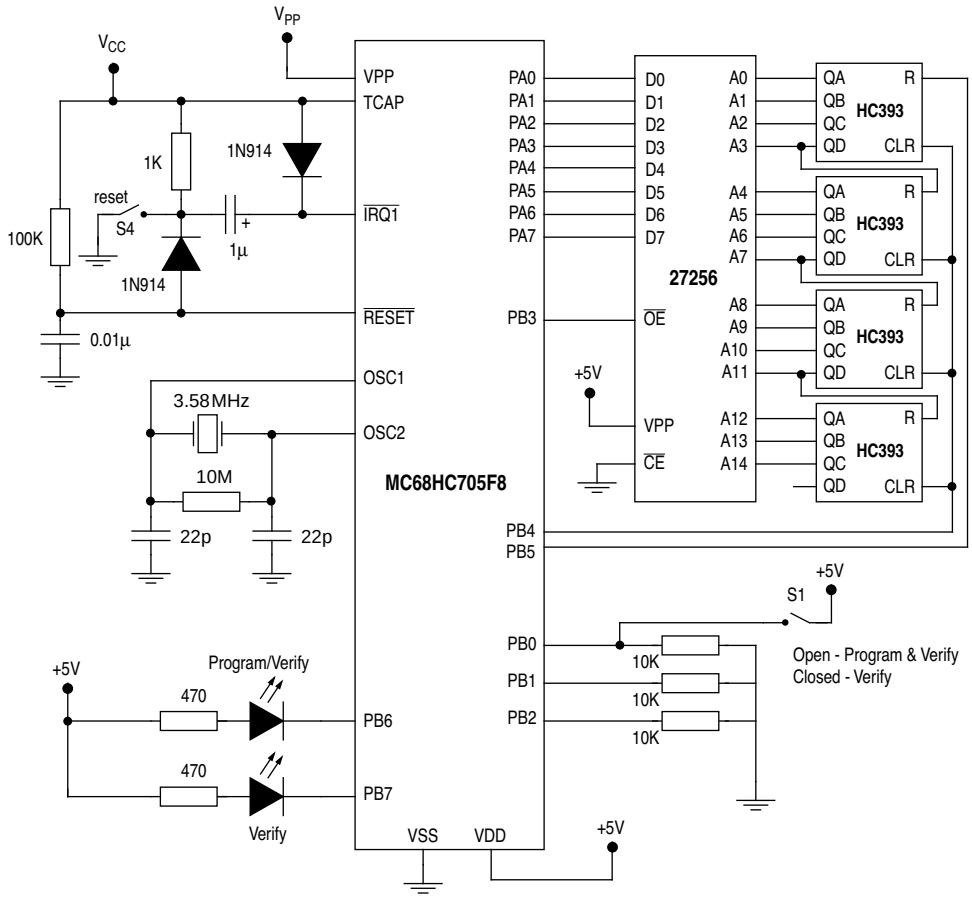


Figure 12-5 EPROM Programming Circuit for Bootstrap Mode

12.3.4 Verify EPROM Contents

The Verify EPROM Contents routine is normally entered automatically after the EPROM is programmed. Direct entry of this mode will cause the EPROM contents to be compared to external memory contents residing at the same address locations. During execution of the verification routine, the Program/Verify LED is illuminated. Upon completion of the verification routine (every location verified) the Program/Verify LED is turned off and the Verified LED is turned on. If the Program/Verify LED does not illuminate, a discrepancy has been detected and the error address location will be placed on the external memory address bus.

THIS PAGE LEFT BLANK INTENTIONALLY

13
ELECTRICAL SPECIFICATIONS

This section contains the electrical specifications for the MC68HC05F8.

13.1 Maximum Ratings

Voltages referenced to V_{SS}

RATINGS	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
$\overline{IRQ}$	V_{in}	V_{SS}–0.3 to 2xV_{DD}+0.3	V
Current Drain per pin excluding V_{DD} and V_{SS}	I_D	25	mA
Operating Temperature	T_A	0 to 70	°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 should be taken to avoid application of any voltage higher than the maximum rated voltages to this high impedance circuit. For proper operation it is recommended that Vin and Vout be constrained to the range $V_{SS} \leq (V_{in} \text{ or } V_{out}) \leq V_{DD}$. Reliability of operation is enhanced if unused inputs are connected to an appropriate logic voltage level (e.g. either V_{SS} or V_{DD}).

13.2 Thermal Characteristics

CHARACTERISTICS	SYMBOL	VALUE	UNIT
Thermal resistance			
- Plastic 56-pin SDIP package	θ_{JA}	50	°C/W
- Plastic 64-pin QFP package	θ_{JA}	50	°C/W

13.3 DC Electrical Characteristics

Table 13-1 DC Electrical Characteristics for 5V Operation

V_{DD}=5.0Vdc ±10%, V_{SS}=0Vdc, temperature range=0 to 70°C

CHARACTERISTICS	SYMBOL	MINIMUM	TYPICAL	MAXIMUM	UNIT
Output voltage $I_{LOAD} \leq -10\mu A$ $I_{LOAD} \leq +10\mu A$	 V_{OH} V_{OL}	 V_{DD}-0.1 –	 – –	 – 0.1	 V V
Output high voltage (I_{LOAD}=1.6mA) PA4-PA5, PB0-PB7, PC0-PC7, PD0-PD7, TCMP, TNX	V_{OH}	V_{DD}-0.8	–	–	V
Output low voltage (I_{LOAD}=1.6mA) PA4-PA5, PB0-PB7, PC0-PC7, PD0-PD7, TCMP, TNX	V_{OL}	–	–	0.4	V
Input high voltage PA0-PA5, PB0-PB7, PC0-PC7, PD0-PD7, TCAP, $\overline{IRQ1}$, $\overline{IRQ2}$, $\overline{RESET}$, OSC1	V_{IH}	0.7xV_{DD}	–	V_{DD}	V
Input low voltage PA0-PA5, PB0-PB7, PC0-PC7, PD0-PD7, TCAP, $\overline{IRQ1}$, $\overline{IRQ2}$, $\overline{RESET}$, OSC1	V_{IL}	V_{SS}	–	0.2xV_{DD}	V
Data Retention Mode	V_{RM}	2.0	–	–	V
Supply current Run Wait Stop	I_{DD}	–	 3.8 1.1 0.9	 5 2 1.5	 mA mA µA
I/O ports high-Z leakage current PA0-PA5, PB0-PB7, PC0-PC7, PD0-PD7	I_{IL}	–	–	±10	µA
Input current TCAP, $\overline{IRQ1}$, $\overline{IRQ2}$, $\overline{RESET}$, DECOIN, OSC1	I_{IN}	–	–	±1	µA
Capacitance ports (as input or output), $\overline{RESET}$, $\overline{IRQ1}$, $\overline{IRQ2}$, TCAP, OSC1	 C_{OUT} C_{IN}	 – –	 – –	 12 8	 pF
Port C high current sinking capability (for 1V saturation)	I_{SK}	–	10	–	mA

13

Table 13-2 DC Electrical Characteristics for 2.7V Operation

V_{DD}=2.7Vdc ±10%, V_{SS}=0Vdc, temperature range=0 to 70°C

CHARACTERISTICS	SYMBOL	MINIMUM	TYPICAL	MAXIMUM	UNIT
Output voltage $I_{LOAD} \leq -10\mu A$ $I_{LOAD} \leq +10\mu A$	 V_{OH} V_{OL}	 V_{DD}-0.1 –	 – –	 – 0.1	 V V
Output high voltage (I_{LOAD}=1.6mA) PA4-PA5, PB0-PB7, PC0-PC7, PD0-PD7, TCMP, TNX	V_{OH}	V_{DD}-0.3	–	–	V
Output low voltage (I_{LOAD}=1.6mA) PA4-PA5, PB0-PB7, PC0-PC7, PD0-PD7, TCMP, TNX	V_{OL}	–	–	0.3	V
Input high voltage PA0-PA5, PB0-PB7, PC0-PC7, PD0-PD7, TCAP, $\overline{IRQ1}$, $\overline{IRQ2}$, $\overline{RESET}$, OSC1	V_{IH}	0.7xV_{DD}	–	V_{DD}	V
Input low voltage PA0-PA5, PB0-PB7, PC0-PC7, PD0-PD7, TCAP, $\overline{IRQ1}$, $\overline{IRQ2}$, $\overline{RESET}$, OSC1	V_{IL}	V_{SS}	–	0.2xV_{DD}	V
Data Retention Mode	V_{RM}	2.0	–	–	V
Supply current Run Wait Stop	I_{DD}	–	 1.5 450 360	 TBD TBD TBD	 mA µA nA
I/O ports high-Z leakage current PA0-PA5, PB0-PB7, PC0-PC7, PD0-PD7	I_{IL}	–	–	±10	µA
Input current TCAP, $\overline{IRQ1}$, $\overline{IRQ2}$, $\overline{RESET}$, DECOIN, OSC1	I_{IN}	–	–	±1	µA
Capacitance ports (as input or output), $\overline{RESET}$, $\overline{IRQ1}$, $\overline{IRQ2}$, TCAP, OSC1	 C_{OUT} C_{IN}	 – –	 – –	 12 8	 pF
Port C high current sinking capability (for 1V saturation)	I_{SK}	–	10	–	mA

13

13.4 DTMF/Melody Generator Electrical Characteristics

Table 13-3 Electrical Specification of sine wave tones at TONEOUT output (including DTMF)

CHARACTERISTICS	MINIMUM	TYPICAL	MAXIMUM	UNIT
Operating voltage	2.5	–	5.5	V
Tone output level:				
Low group - row	0.125	0.15	0.16	V_{RMS}
High group - column	0.158	0.192	0.205	V_{RMS}
Frequency deviation (DTMF)	–0.65	–	+0.65	%
Frequency deviation (Melody)	–1.5	–	+1.5	%
Tone output DC level	0.45	0.50	0.55	V_{DD}
High group pre-emphasis	1	2	3	dB

Table 13-4 Electrical Specification of square wave tones at TONEOUT output

CHARACTERISTICS	MINIMUM	TYPICAL	MAXIMUM	UNIT
Operating voltage	2.5	–	5.5	V
Tone output level:				
Low group - row	0.19	0.21	0.24	V_{P-P}
High group - column	0.24	0.27	0.30	V_{P-P}
Frequency deviation (Melody)	–1.5	–	+1.5	%
Tone output DC level (+0.5 V_{P-P} value)	0.45	0.50	0.55	V_{DD}

Table 13-5 Electrical Specification of ToneX at TONEX output

CHARACTERISTICS	MINIMUM	TYPICAL	MAXIMUM	UNIT
Tone output level (square wave)	–	V_{DD}	–	V_{P-P}
Frequency deviation	–1.5	–	1.5	%
Tone output DC level	0.45	0.50	0.55	V_{DD}

13.5 Control Timing

Table 13-6 Control Timing for 5V Operation

(V_{DD}=5.0Vdc ±10%, V_{SS}=0Vdc, temperature range=0 to 70°C)

CHARACTERISTICS	SYMBOL	MINIMUM	MAXIMUM	UNIT
Frequency of operation Crystal option External clock option	f_{OSC}	– dc	3.58 3.58	MHz MHz
Internal operating frequency (f_{OSC}/2) Crystal External clock	 f_{OP} f_{OP}	 – dc	 1.8 1.8	 MHz MHz
Processor cycle time	t_{CYC}	556	–	ns
Crystal oscillator start-up time	t_{OXOV}	–	100	ms
Stop recovery start-up time (crystal oscillator)	t_{ILCH}		100	ms
External RESET pulse width	t_{RL}	1.5	–	t_{CYC}
Power-on RESET output pulse width 4064 cycle 16 cycle	 t_{PORL} t_{PORL}	 4064 16	 – –	 t_{CYC} t_{CYC}
Watchdog RESET output pulse width	t_{DOGL}	1.5	–	t_{CYC}
Watchdog time-out	t_{DOG}	0.25	4	sec
Timer A Resolution(1) Input capture pulse width Input capture pulse period	 t_{ARESL} t_{TH}, t_{TL} t_{TLTL}	 4 125 –(2)	 64 – –	 t_{CYC} ns t_{CYC}
Timer B Resolution	 t_{BRESL}	 2.25	 18	 µs
Interrupt pulse width (edge-triggered)	t_{ILIH}	125	–	ns
Interrupt pulse period	t_{ILIL}	–(3)	–	t_{CYC}

(1) Since a 2-bit prescaler in the timer must count four external cycles (t_{CYC}), this is the limiting factor in determining the timer resolution.

(2) The minimum period t_{TLTL} should not be less than the number of cycle times it takes to execute the capture interrupt service routine plus 24 t_{CYC}.

(3) 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}.

Table 13-7 Control Timing for 2.7V Operation

(V_{DD}=3.3Vdc ±10%, V_{SS}=0Vdc, temperature range=0 to 70°C)

CHARACTERISTICS	SYMBOL	MINIMUM	MAXIMUM	UNIT
Frequency of operation Crystal option External clock option	f_{OSC}	– dc	3.58 3.58	MHz MHz
Internal operating frequency (f_{OSC}/2) Crystal External clock	 f_{OP} f_{OP}	 – dc	 1.8 1.8	 MHz MHz
Processor cycle time	t_{CYC}	556	–	ns
Crystal oscillator start-up time	t_{OXOV}	–	100	ms
Stop recovery start-up time (crystal oscillator)	t_{ILCH}		100	ms
External RESET pulse width	t_{RL}	1.5	–	t_{CYC}
Power-on RESET output pulse width 4064 cycle 16 cycle	 t_{PORL} t_{PORL}	 4064 16	 – –	 t_{CYC} t_{CYC}
Watchdog RESET output pulse width	t_{DOGL}	1.5	–	t_{CYC}
Watchdog time-out	t_{DOG}	0.25	4	sec
Timer A Resolution(1) Input capture pulse width Input capture pulse period	 t_{ARESL} t_{TH}, t_{TL} t_{TLTL}	 4 250 –(2)	 64 – –	 t_{CYC} ns t_{CYC}
Timer B Resolution	 t_{BRESL}	 2.25	 18	 µs
Interrupt pulse width (edge-triggered)	t_{ILIH}	250	–	ns
Interrupt pulse period	t_{ILIL}	–(3)	–	t_{CYC}

(1) Since a 2-bit prescaler in the timer must count four external cycles (t_{CYC}), this is the limiting factor in determining the timer resolution.

(2) The minimum period t_{TLTL} should not be less than the number of cycle times it takes to execute the capture interrupt service routine plus 24 t_{CYC}.

(3) 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}.

13.6 *Programming Operation Electrical Characteristics*

V_{DD}=5.0Vdc ±5%, V_{SS}=0Vdc, temperature range=20 to 30°C)

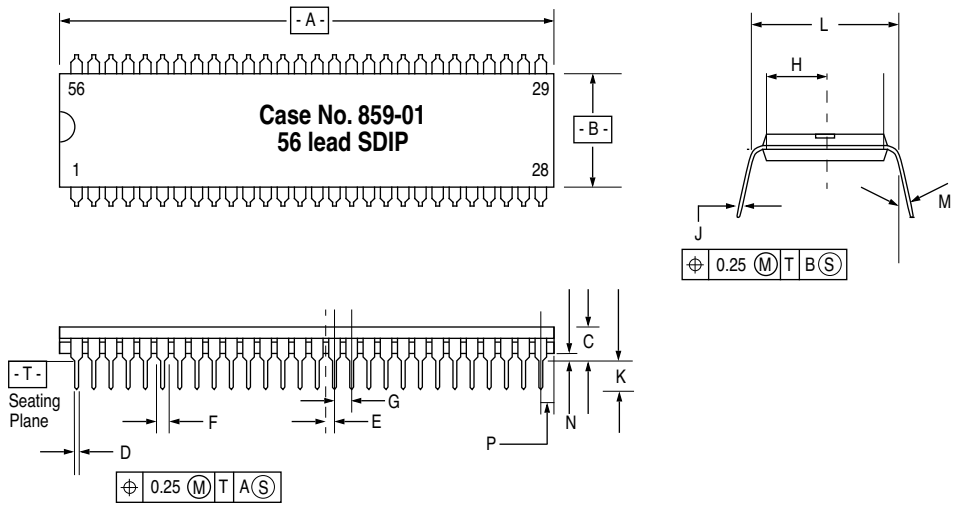
CHARACTERISTICS	*SYMBOL*	*MINIMUM*	*TYPICAL*	*MAXIMUM*	*UNIT*
Programming voltage	V_{PP}	*13*	*14*	*15*	*V*
V_{PP} supply current *$V_{PP}=V_{DD}$* *V_{PP}=14V*	I_{PP}	*3* *–*	*4* *4*	*5* *8*	*µA* *mA*
Programming bus frequency	F_{BUS}	*1.70*	*1.79*	*1.89*	*MHz*
Bootstrap programming mode voltage (IRQ1 pin, I_{IN}=100µA max.)	V_{IHTP}	*9.0*	*10.0*	*10.3*	*V*

THIS PAGE LEFT BLANK INTENTIONALLY

14
MECHANICAL SPECIFICATIONS

This section provides the mechanical dimension for the 56-pin SDIP and 64-pin QFP packages for the MC68HC05F8.

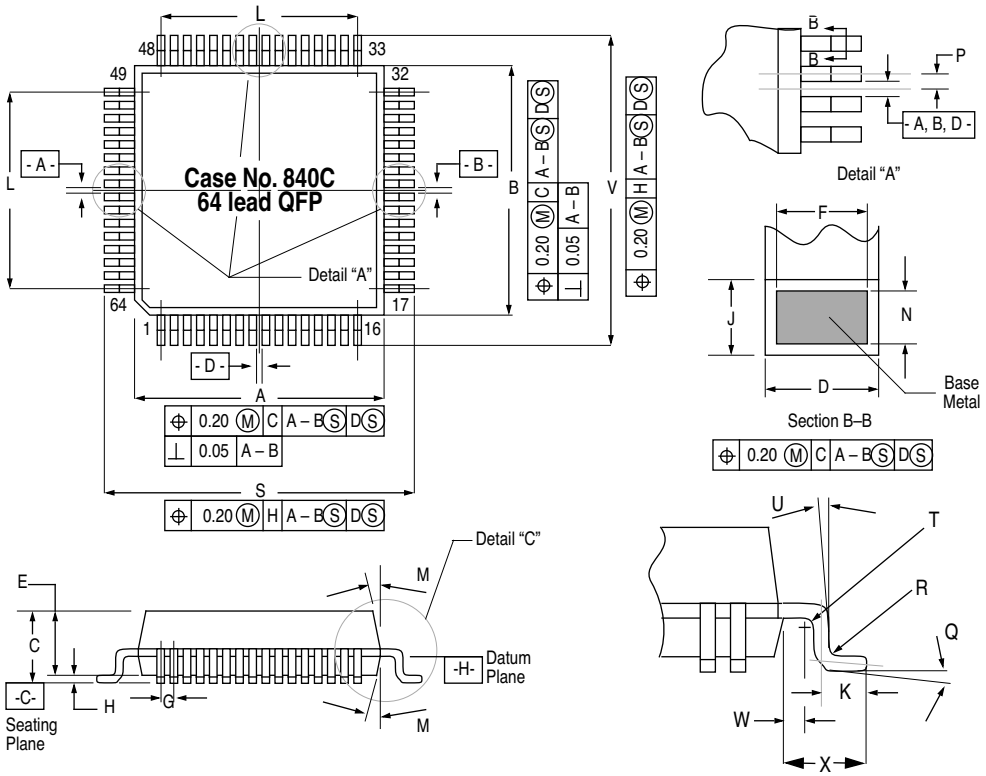
14.1 56-pin SDIP Package



Dim.	Min.	Max.	Notes	Dim.	Min.	Max.
A	51.69	52.45	1. Dimensions and tolerancing per ANSI Y 14.5 1982. 2. All dimensions in mm. 3. Dimension L to centre of lead when formed parallel. 4. Dimensions A and B do not include mould flash. Allowable mould flash is 0.25 mm.	H	7.62 BSC	
B	13.72	14.22		J	0.20	0.38
C	3.94	5.08		K	2.92	3.43
D	0.36	0.56		L	15.24 BSC	
E	0.89 BSC			M	0°	15°
F	0.81	1.17		N	0.51	1.02
G	1.778 BSC			P	1.78	2.29

Figure 14-1 56-pin SDIP Mechanical Dimensions

14.2 64-pin QFP Package



Dim.	Min.	Max.
A	13.90	14.10
B	13.90	14.10
C	2.067	2.457
D	0.30	0.45
E	2.00	2.40
F	0.30	—
G	0.80 BSC	
H	0.067	0.250
J	0.130	0.230
K	0.50	0.66
L	12.00 REF	

Notes

1. Datum Plane –H– is located at bottom of lead and is coincident with the lead where the lead exits the plastic body at the bottom of the parting line.
2. Datums –A–, –B– and –D– to be determined at Datum Plane –H–.
3. Dimensions S and V to be determined at seating plane –C–.
4. Dimensions A and B do not include mould protrusion. Allowable mould protrusion is 0.25 mm per side. Dimensions A and B do include mould mismatch and are determined at Datum Plane –H–.
5. Dimension D does not include dambar protrusion. Allowable dambar protrusion shall be 0.08 total in excess of the D dimension at maximum material condition. Dambar cannot be located on the lower radius or the foot.
6. Dimensions and tolerancing per ANSI Y 14.5M, 1982.
7. All dimensions in mm.

Dim.	Min.	Max.
M	5°	10°
N	0.130	0.170
P	0.40 BSC	
Q	2°	8°
R	0.13	0.30
S	16.20	16.60
T	0.20 REF	
U	9°	15°
V	16.20	16.60
W	0.042 NOM	
X	1.10	1.30

Figure 14-2 64-pin QFP Mechanical Dimensions

How to reach us:

MFAX: RMFAX0@email.sps.mot.com – TOUCHTONE (602) 244-6609

INTERNET: http://Design-NET.com

USA/EUROPE: Motorola Literature Distribution; P.O. Box 20912; Phoenix, Arizona 85036. 1-800-441-2447

JAPAN: Nippon Motorola Ltd.; Tatsumi-SPD-JLDC, Toshikatsu Otsuki, 6F Seibu-Butsuryu-Center, 3-14-2 Tatsumi Koto-Ku, Tokyo 135, Japan. 03-3521-8315

HONG KONG: Motorola Semiconductors H.K. Ltd.; 8B Tai Ping Industrial Park, 51 Ting Kok Road, Tai Po, N.T., Hong Kong. 852-26629298

MOTOROLA

MC68HC05F8D/H