MC68HC05G1

HC05

MC68HC05G1
MC68HC705G1

TECHNICAL
DATA

TECHNICAL DATA

MOTOROLA

MOTOROLA

MC68HC05G1
MC68HC705G1

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

CUSTOMER FEEDBACK QUESTIONNAIRE (MC68HC05G1D/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	CLOCKS	☐	☐	☐	☐	☐
SECTION 7	PROGRAMMABLE TIMER	☐	☐	☐	☐	☐
SECTION 8	SERIAL PERIPHERAL INTERFACE	☐	☐	☐	☐	☐
SECTION 9	ANALOG TO DIGITAL CONVERTER	☐	☐	☐	☐	☐
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: MC68HC05G1D/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
CLOCKS

7
PROGRAMMABLE TIMER

8
SERIAL PERIPHERAL INTERFACE

9
ANALOG TO DIGITAL CONVERTER

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 MC68HC05G1 HCMOS microcontroller is a member of the M68HC05 family of low-cost single-chip microcontrollers. This 8-bit microcontroller unit (MCU) contains two on-chip oscillators, CPU, RAM, ROM, I/O, timer, serial peripheral interface, real time clock, alarm, A to D converter, and watchdog. The MC68HC05G1 is available in 56-pin SDIP and 64-pin QFP packages.

The MC68HC705G1 is an EPROM version of the MC68HC05G1. All references to the MC68HC05G1 apply equally to the MC68HC705G1, unless otherwise stated. *References to the MC68HC705G1 are italicized in the text.*

1.1 Features

- 8-bit architecture
- Power saving Stop, Wait modes
- 176 bytes of on-chip RAM (including 64 bytes for stack)
- 8K bytes of on-chip ROM; *11776 bytes EPROM for MC68HC705G1*
- 40 bidirectional I/O lines and 8 fixed input lines
- One fixed, and 2 general purpose external interrupts
- Real time clock (RTC)
- Internal 16-bit counter
- Serial Peripheral Interface (SPI)
- Four channel 8-bit A to D converter
- COP Watchdog counter
- Self-check mode
- Available in 56-pin SDIP and 64-pin QFP packages

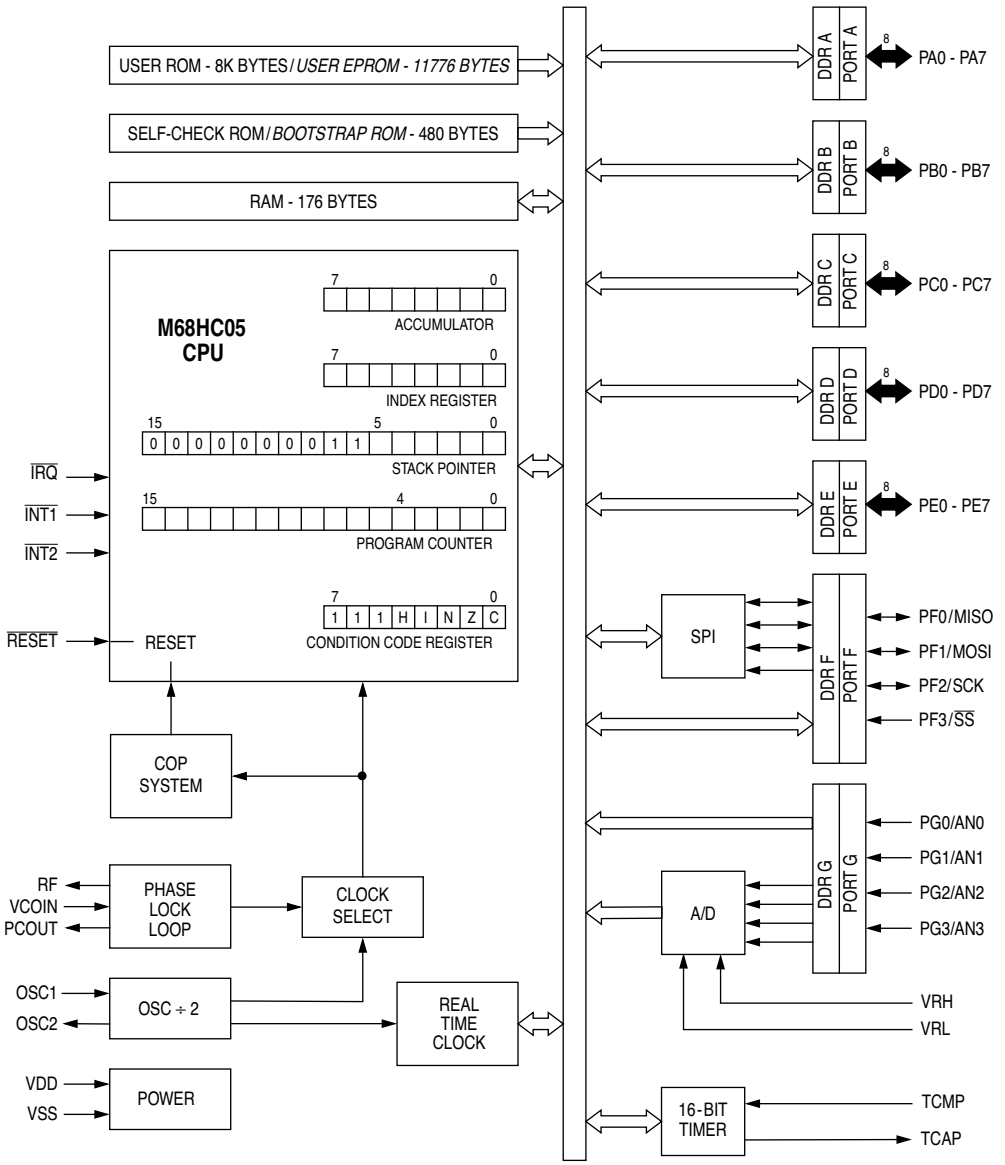


Figure 1-1 MC68HC05G1/*MC68HC705G1* Block Diagram

2
PIN DESCRIPTIONS

This section provides a description of the functional pins and I/O programming of the MC68HC05G1 microcontroller.

2.1 Functional Pin Descriptions

PIN NAME	56-pin SDIP PIN No.	64-pin QFP PIN No.	DESCRIPTION
VDD, VSS	39, 29	37, 25	Power is supplied to the MCU using these two pins. VDD is power and VSS is ground.
$\overline{IRQ}$	35	32	$\overline{IRQ}$ is 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.
$\overline{INT1}$, $\overline{INT2}$	34, 32	31, 29	$\overline{INT1}$ & $\overline{INT2}$ are general purpose external interrupt input pins. They are accompanied with their individual enable, flag, and status bits. They 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.
$\overline{RESET}$	36	34	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	16	11	The TCAP input controls the input capture feature for the on-chip programmable timer.
TCMP	15	10	The TCMP pin provides an output for the output compare feature of the on-chip programmable timer.
OSC1, OSC2	37, 38	35, 36	These pins provide connections to the on-chip oscillator. The crystal frequency is 32.768 KHz. OSC1 may be driven by an external oscillator if an external crystal circuit is not used.
PA0-PA7	47-54	45, 46, 48-52, 54	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.

PIN NAME	56-pin SDIP PIN No.	64-pin QFP PIN No.	DESCRIPTION
PB0-PB7	17-24	12-19	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	55, 56, 1-6	55-60, 62, 63	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.
PD0-PD7	14-7	9-5, 3, 2, 64	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.
PE0-PE7	Not bonded out in the 56-pin package	47, 53, 61, 1, 5, 21, 26, 33	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. Port E is only available for the 64-pin QFP package; to avoid leakage current, these eight lines should be programmed to output modes for the 56-pin package.
PF0/MISO PF1/MOSI PF2/SCK PF3/$\overline{SS}$	25 26 27 28	20 22 23 24	These four lines comprise of the fixed input port F. It is the default setting at power-on or reset. PF0-PF3 becomes MISO, MOSI, SCK, & $\overline{SS}$ respectively when the Serial Peripheral Interface is activated by setting SPE of the Serial Peripheral Control register (bit 6 of address $2A).
PG0/AN0 PG1/AN1 PG2/AN2 PG3/AN3	42 43 44 45	40 41 42 43	These four lines comprise of the fixed input port G. It is the default setting at power-on or reset. If the ADON bit of the A/D Control & Status register (bit 5 of address $29) is set, PG0-PG3 become AN0-AN3, analog inputs for the on-chip A/D converter.
VRH, VRL	40, 41	38, 39	The VRH input pin is the high reference voltage for the A/D converter. The VRL input pin is the low reference voltage for the A/D converter.
NC/*VPP*	46	44	This pin is used as the programming voltage pin for the EPROM version, *MC68HC705G1*. It is connected to VDD for normal operation. This pin is not used in the standard ROM part, MC68HC05G1.
RF, VCOIN, PCOUT	30, 33, 31	27, 30, 28	These pins provide connection to the on-chip phase lock loop (PLL) oscillator.

2.2 Pin Assignments

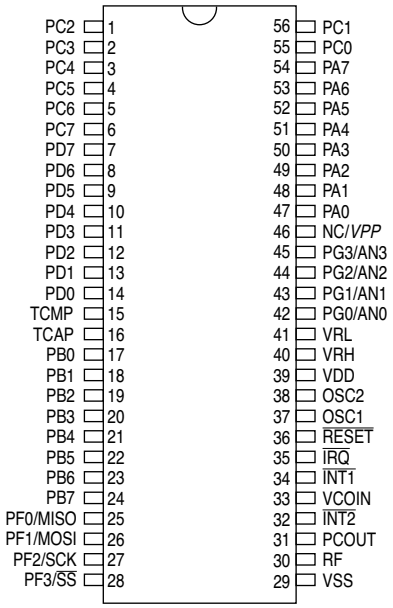


Figure 2-1 Pin Assignments for 56-pin SDIP package

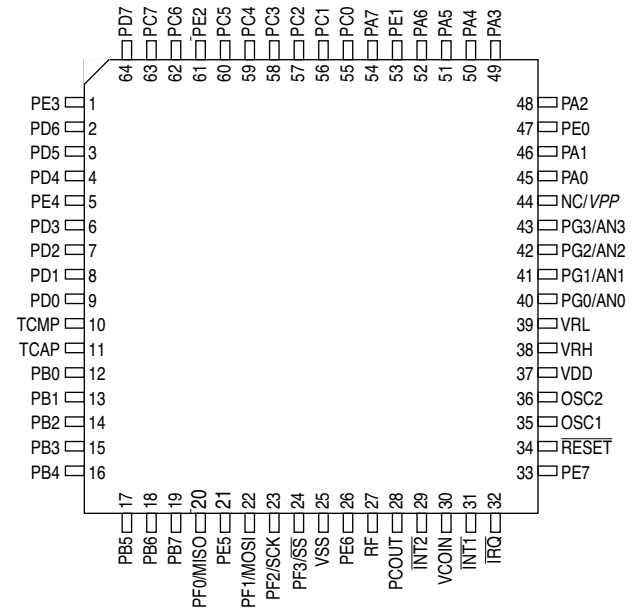


Figure 2-2 Pin Assignments for 64-pin QFP package

2.3 Input/Output Programming

2

2.3.1 Parallel Ports

Port A, B, C, D, and port E 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 has an associated 8-bit Data Direction register. Any port A, B, C, D or port E 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 and port E pins as inputs. The data direction registers are capable of being written to or read by the CPU. 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 Fixed Ports

Port F is a 4-bit fixed input port that continually monitors the external pins whenever the SPI system is disabled. During power-on reset or external reset all four bits become valid input ports because all special function output drivers are disabled. For example, with the serial peripheral interface (SPI) system disabled (SPE=0) PF0 through PF3 will read the state of the pin at the time of the read operation. No data direction register is associated with the port when it is used as an input.

Port G is a fixed, input-only, 4-bit port, which can be read at any time; it reads the four analogue inputs to the A/D converter, when it is enabled. Port G can still be read during an A/D conversion sequence, but this may inject noise on the analogue inputs, resulting in reduced accuracy of the A/D. Furthermore, performing a digital read of port G with levels other than V_{DD} or V_{SS} on the port G pins will result in greater power dissipation during the read cycle. Note that as port G is an input only port there is no Data Direction Register (DDR) associated with it. Also, at power-on or external reset, the A/D converter is disabled.

Note: It is recommended that all unused inputs should be tied to an appropriate logic level (e.g. either V_{DD} or V_{SS}).

2.3.3 Serial Port (SPI)

The serial peripheral interface (SPI) uses the port F pins for its function. The SPI function requires four of the pins (PF0-PF3) for its Master In Slave Out (MISO), Master Out Slave In (MOSI), System Clock (SCK), and Slave Select ($\overline{SS}$) respectively.

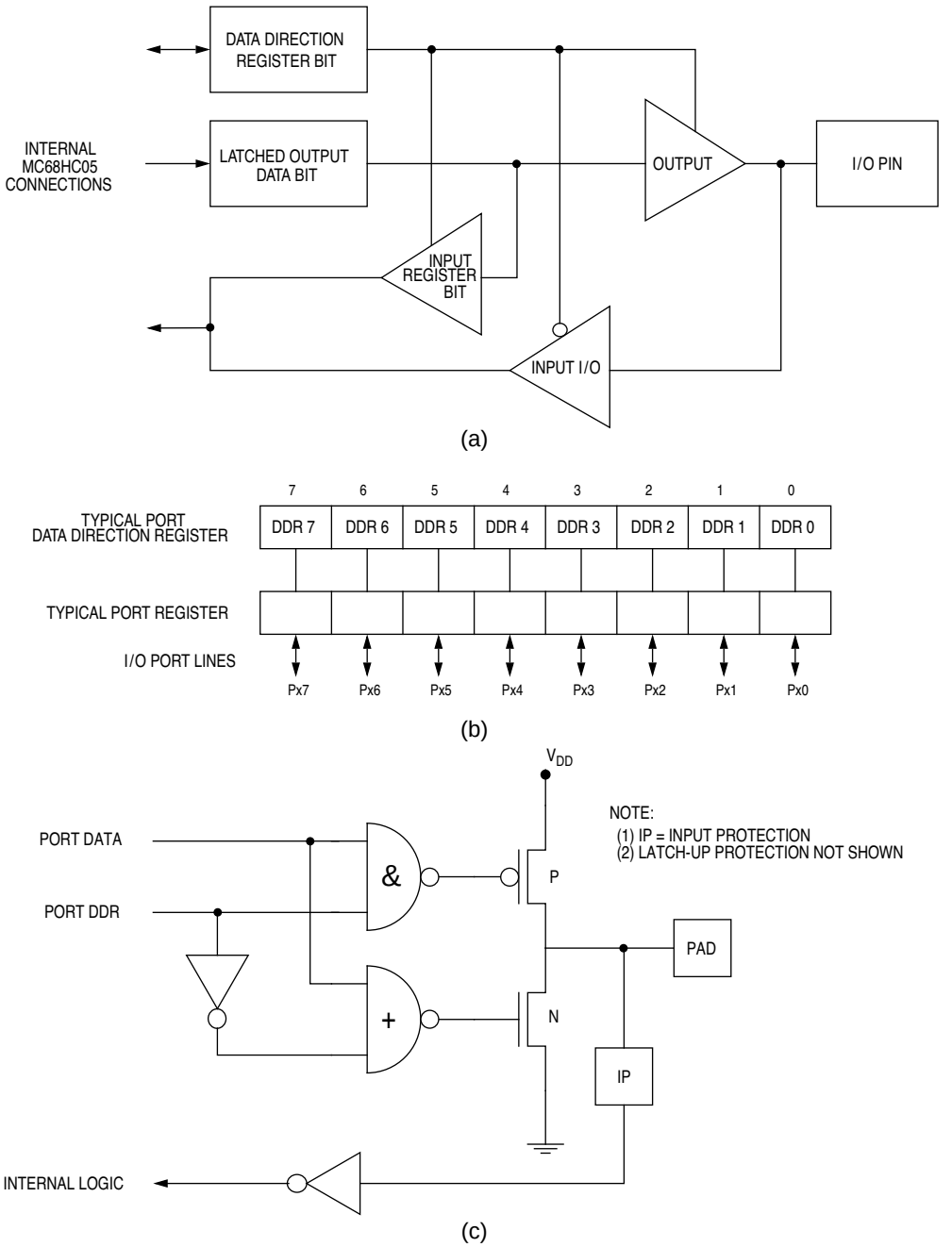


Figure 2-3 Parallel Port I/O Circuitry

3
MEMORY AND REGISTERS

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

3.1 Memory Map

The CPU can address 16K-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 MC68HC05G1/*MC68HC705G1*.

3.2 Input/Output Section

The first 48 addresses of memory space, $0000-$002F, are the I/O section. These are the addresses of the I/O control registers, status registers, and data registers.

3.3 RAM

The 176 addresses from $0050-$00FF are RAM locations. The CPU uses the top 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.

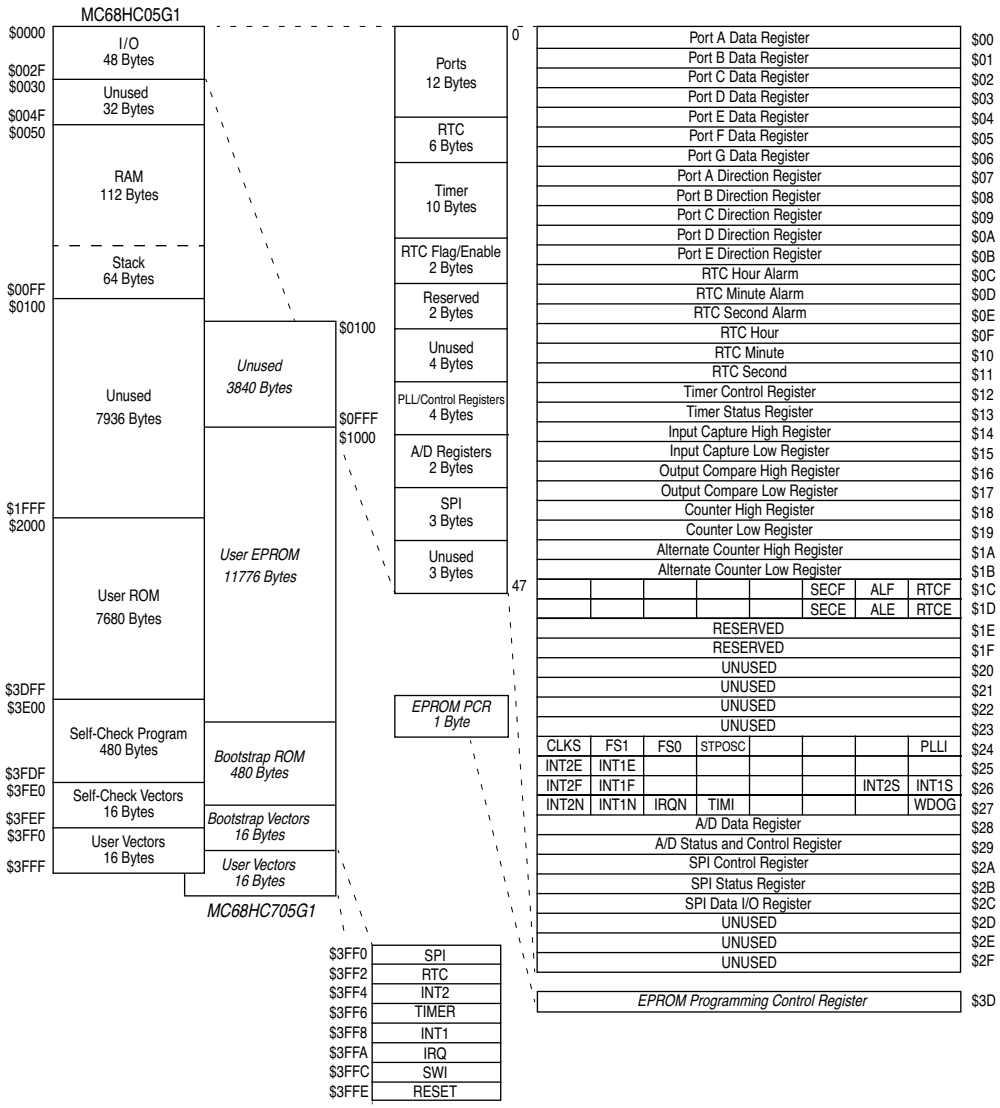


Figure 3-1 MC68HC05G1/*MC68HC705G1* Memory Map

Table 3-1 MC68HC05G1/*MC68HC705G1* I/O 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 3	bit 2	bit 1	bit 0
$06	Port G data					bit 3	bit 2	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	RTC hour alarm								
$0D	RTC minute alarm								
$0E	RTC second alarm								
$0F	RTC hour								
$10	RTC minute								
$11	RTC second								
$12	Timer control	ICIE	OCIE	TOIE				IEDG	OLVL
$13	Timer status	ICF	OCF	TOF					
$14	Input capture high								
$15	Input capture low								
$16	Output compare high								
$17	Output compare low								
$18	Counter high								
$19	Counter low								
$1A	Alternate counter high								
$1B	Alternate counter low								
$1C	RTC status						SECF	ALF	RTCF
$1D	RTC control						SECE	ALE	RTCE
$1E	Reserved								
$1F	Reserved								

Table 3-1 MC68HC05G1/*MC68HC705G1* I/O Registers

Address	Register Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
$20	Unused								
$21	Unused								
$22	Unused								
$23	Unused								
$24	PLL status and control	CLKS	FS1	FS0	STPOSC				PLLI
$25	External interrupt control	INT2E	INT1E						
$26	External interrupt status	INT2F	INT1F					INT2S	INT1S
$27	Miscellaneous control	INT2N	INT1N	IRQN	TIMI				WDOG
$28	A/D data								
$29	A/D status and control	COCO	ADRC	ADON		CH3	CH2	CH1	CH0
$2A	SPI control	SPIE	SPE		MSTR	CPOL	CPHA	SPR1	SPR0
$2B	SPI status	SPIF	WCOL		MODF				
$2C	SPI data I/O								
$2D	Unused								
$2E	Unused								
$2F	Unused								

$3D	*EPROM programming control*							*ELAT*	*PGM*

4
RESETS

The MC68HC05G1 can be reset in three ways: by the power-on-reset function, by an active low input to the $\overline{\text{RESET}}$ pin, and by a COP watchdog timer reset, if the watchdog timer is enabled.

4.1 Power-On Reset (POR) Pin

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.

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 COP Watchdog Reset

The COP (Computer Operating Properly) watchdog timer function is implemented on this device by using the output of the COP counter. The minimum COP reset rates are shown in Table 4-1. If the COP circuit times out, an internal reset is generated and the reset vector is fetched (at $3FFE & $3FFF). Preventing a COP time-out is achieved by writing a '0' to bit 0 of address $3FF0. The COP counter has to be cleared periodically by software with a period less than COP reset rate.

COP counter will stop counting in Wait and Stop modes. Counting continues when it wakes up from WAIT mode, and a 4064 cycle delay after waking up from Stop mode.

The watchdog counter system is controlled by the WDOG bit in the Miscellaneous Control register (bit 0 of address $27). After power-on or external reset the watchdog system is disabled. Writing a "1" to the WDOG bit will enable the watchdog system and the counter starts counting. Once enabled, the watchdog system cannot be disabled by software. Writing a "0" to bit 0 of address $3FF0 will reset watchdog counter to prevent a watchdog time-out.

Note: The COP watchdog reset rate is 2^{20} times the CPU clock cycle.

Table 4-1 COP Reset Rate

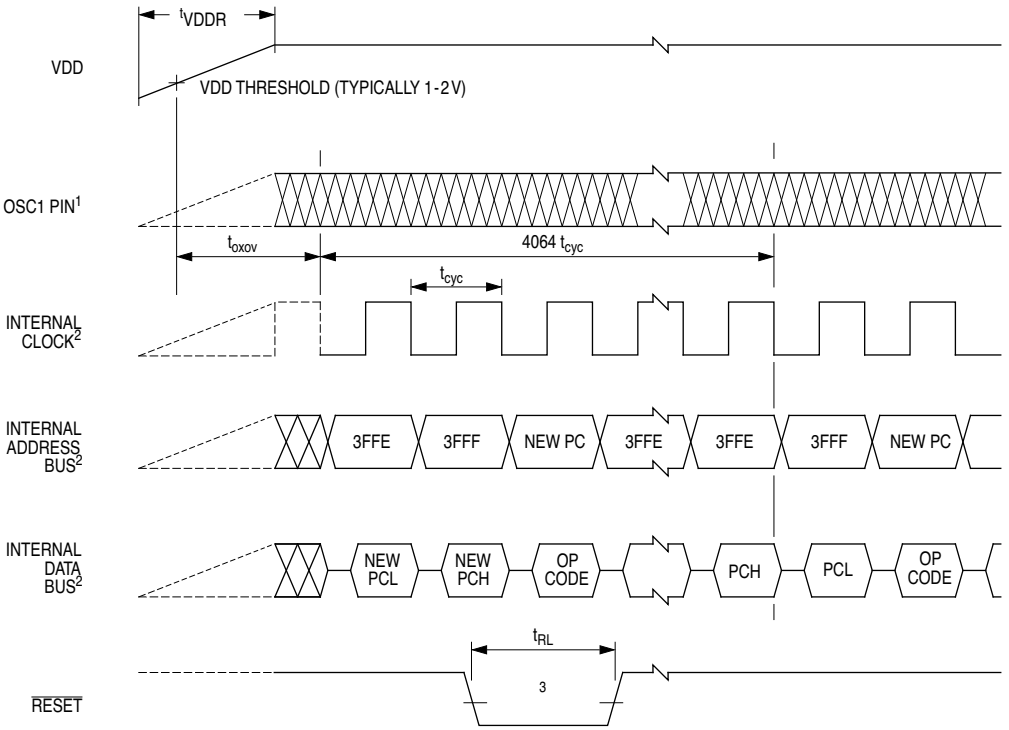
FS1	FS0	CPU Bus Rate	Minimum COP Reset
0	0	0.164 MHz	6.4 s
0	1	1.311 MHz	800 ms
1	0	0.328 MHz	3.2 s
1	1	2.622 MHz	400 ms
FS1 and FS0 are bits in the PLL Status and Control register.			

Table 4-2 shows the internal circuit actions on reset, but not necessary in order of occurrence.

Table 4-2 Reset Action on internal Circuit

	DEFAULT CONDITIONS AFTER RESET
1	Timer prescaler resets to zero state.
2	Timer counter configures to $FFFC.
3	Timer output compare (TCMP) bit resets to zero.
4	All timer interrupt enable bits cleared (ICIE, OCIE, and TOIE) to disable timer interrupts. The OLVL timer bit is also cleared by reset.
5	All data direction registers cleared to zero (default as inputs).
6	Count down register is set to 3.
7	Hours, minutes, and seconds registers are set to zero.
8	Stack pointer configured to $00FF.
9	Internal address bus forced to restart vector ($3FFE-$3FFF).
10	I bit of condition code register set to logic 1.
11	STOP latch cleared.
12	External interrupt latch cleared.
13	WAIT latch cleared.
14	SPI disabled (serial output enable control bit SPE=0). Other SPI bits cleared are SPIE, MSTR, SPIF, WCOL, and MODF.
15	Serial interrupt enable bit cleared (SPIE).
16	SPI system configured to slave mode.
17	RTC interrupt enable bit is cleared. Other RTC bits cleared are RTCF, ALF, and SECF.
18	All bits in address $1C, $1D, $24, $25, $26, and $27 are cleared.
19	A/D disabled
20	Watchdog counter reset.
21	Watchdog WDOG bit cleared.

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

5.1 Introduction

The MC68HC05G1 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, blocks all interrupts except the software interrupt, SWI. Interrupts such as Timer, RTC, and SPI 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 blocks 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 (except RTC, $\overline{\text{IRQ}}$, $\overline{\text{INT1}}$ and $\overline{\text{INT2}}$). 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 and Figure 5-3 shows the program flow 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.

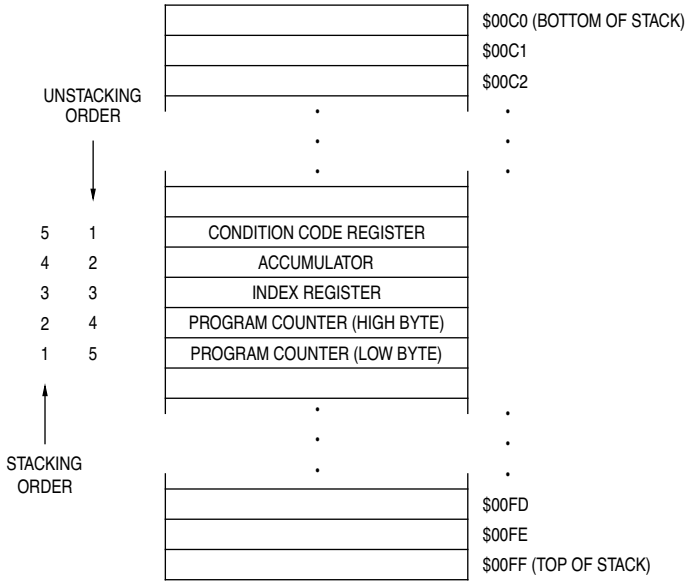


Figure 5-1 Interrupt Stacking Order

Table 5-1 Reset/Interrupt Vector Addresses

Register	Flag Name	Interrupt	CPU Interrupt	Vector Address
–	–	Reset	$\overline{\text{RESET}}$	$3FFE-$3FFF
–	–	Software	SWI	$3FFC-$3FFD
–	–	External Interrupt	$\overline{\text{IRQ}}$	$3FFA-$3FFB
External Interrupt Status	INT1F	External Interrupt	$\overline{\text{INT1}}$	$3FF8-$3FF9
Timer Status	ICF	Input Capture	TIMER	$3FF6-$3FF7
	OCF	Output Compare		
	TOF	Timer Overflow		
External Interrupt Status	INT2F	External Interrupt	$\overline{\text{INT2}}$	$3FF4-$3FF5
RTC Status	SECF	Per Second	RTC	$3FF2-$3FF3
	ALF	Alarm		
	RTCF	Per Day		
SPI Status	SPIF	Data Transfer Complete	SPI	$3FF0-$3FF1
	WCOL	Write Collision		
	MODF	Mode Fault		

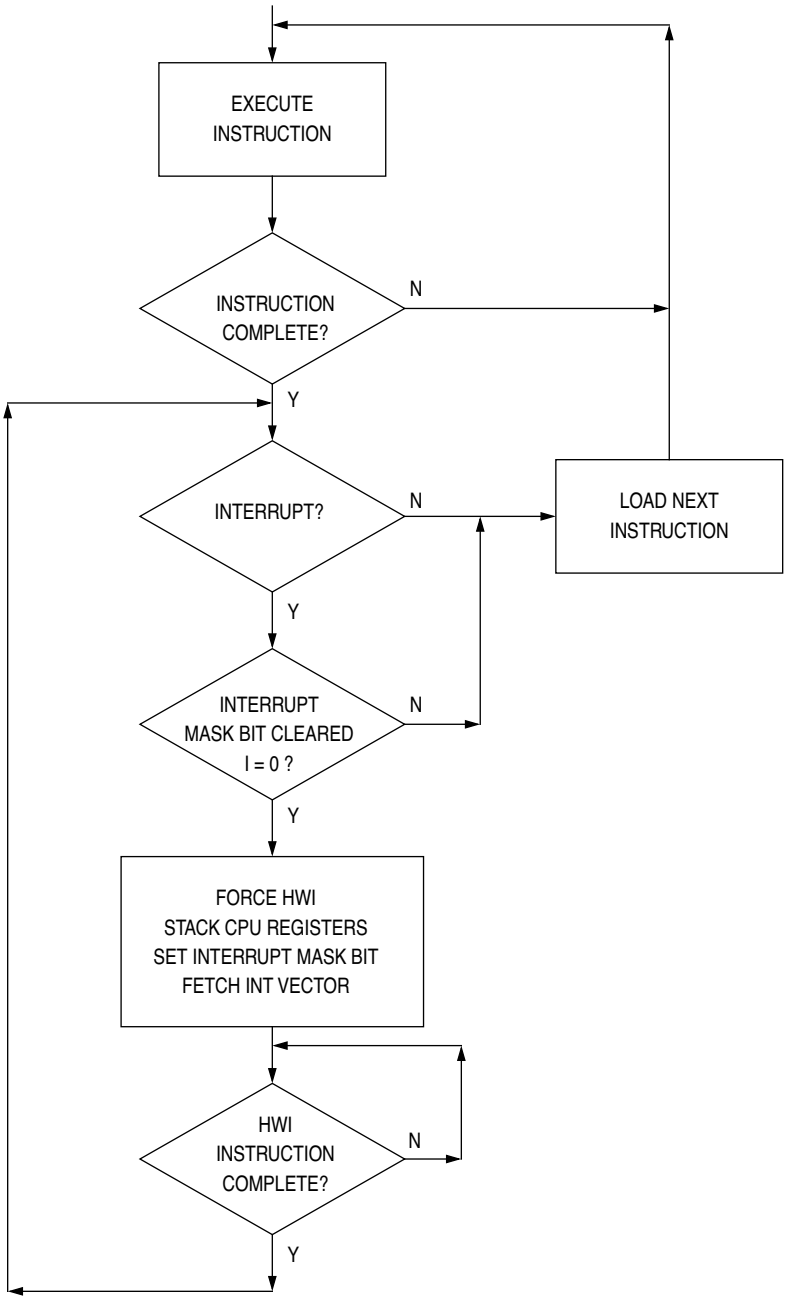


Figure 5-2 $\overline{\text{IRQ}}$ Interrupt Flowchart

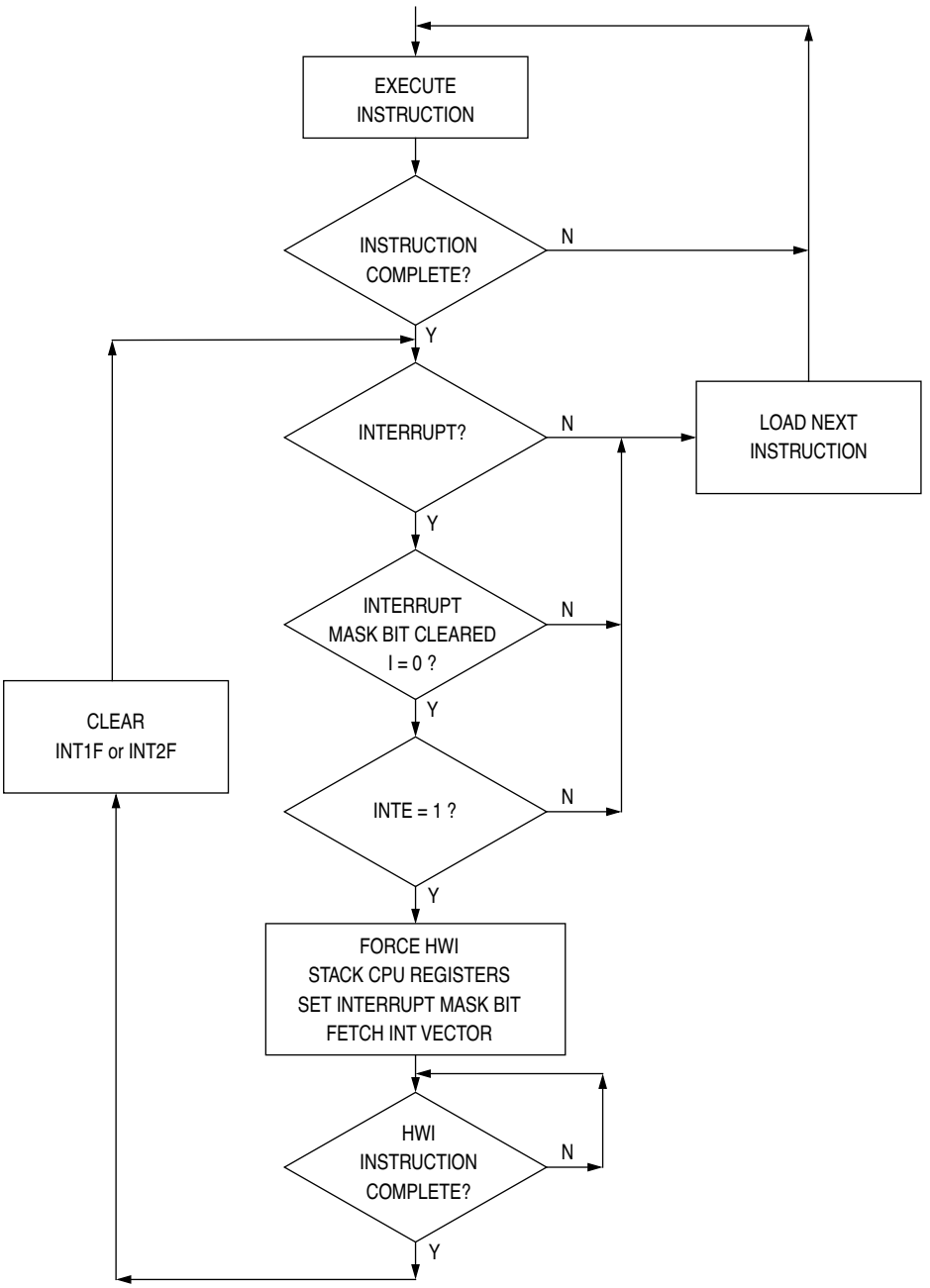


Figure 5-3 $\overline{\text{INT1}}$ and $\overline{\text{INT2}}$ Interrupt Flowchart

5.2 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 $3FFE and $3FFF. The interrupt mask of the Condition Code register is also set. Much of the MCU is configured to some known state as described in Table 4-1.

2) STOP — If the user chooses to use an interrupted RTC, the STOP instruction causes the oscillator to be turned off and the processor "sleeps" until an external interrupt ($\overline{\text{IRQ}}$, $\overline{\text{INT1}}$ or $\overline{\text{INT2}}$) or $\overline{\text{RESET}}$ occurs. If non-interrupted RTC is used, only the internal processor clock is turned off. See section 7 on Low Power Modes.

3) WAIT — The WAIT instruction causes all processor clocks to stop, but leaves the RTC, Timer, and SPI clocks running. This "rest" state of the processor can be cleared by $\overline{\text{RESET}}$, an external interrupt ($\overline{\text{IRQ}}$, $\overline{\text{INT1}}$ or $\overline{\text{INT2}}$), RTC, Timer or SPI interrupt. There are no special wait vectors for these individual interrupts. See section 7 on Low Power Modes.

5.3 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 $3FFC and $3FFD.

5.4 External Interrupts ($\overline{\text{IRQ}}$, $\overline{\text{INT1}}$ & $\overline{\text{INT2}}$)

There are two types of external interrupt in the MC68HC05G1. They can be software configured for negative-edge or level sensitive triggering.

$\overline{\text{IRQ}}$ — When the signal of the external interrupt pin, $\overline{\text{IRQ}}$, satisfies the condition selected by the IRQN bit in the Miscellaneous Control register (bits 5 of address $27) then an external interrupt occurs. 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 $3FFA and $3FFB.

$\overline{\text{INT1}}$ and $\overline{\text{INT2}}$ When the signal of the external interrupt pin, $\overline{\text{INT1}}$ or $\overline{\text{INT2}}$, satisfies the condition selected by the INT1N & INT2N in the Miscellaneous Control register (bits 6 & 7 of address $27) then an external interrupt occurs. 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 $3FF8 & $3FF9 for $\overline{\text{INT1}}$, and $3FF4 & $3FF5 for $\overline{\text{INT2}}$. After servicing the interrupt, flags are cleared by writing a "1" to the corresponding bit; otherwise the CPU will keep servicing the interrupt.

The interrupt logic recognizes negative edge transitions and pulses (special case of negative edges) on the external interrupt lines. Figure 5-4 shows both a block diagram and timing for the interrupt lines ($\overline{\text{IRQ}}$, $\overline{\text{INT1}}$, $\overline{\text{INT2}}$) to the processor. The first method is used if single pulses on the interrupt line is 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 interrupts 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.

5.4.1 External Interrupt Control Register ($25)

Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
$25	INT2E	INT1E							00-- ----

INT1E

1 (set) – External interrupt $\overline{\text{INT1}}$ enable

0 (clear) – External interrupt $\overline{\text{INT1}}$ disable

INT2E

1 (set) – External interrupt $\overline{\text{INT2}}$ enable

0 (clear) – External interrupt $\overline{\text{INT2}}$ disable

5.4.2 External Interrupt Status Register ($26)

Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
$26	INT2F	INT1F					INT2S	INT1S	00-- --uu

INT1S

1 (set) – $\overline{\text{INT1}}$ input pin is high

0 (clear) – $\overline{\text{INT1}}$ input pin is low

INT2S

1 (set) – $\overline{\text{INT2}}$ input pin is high

0 (clear) – $\overline{\text{INT2}}$ input pin is low

INT1F

1 (set) – External interrupt on $\overline{\text{INT1}}$ pin has occurred

0 (clear) – External interrupt on $\overline{\text{INT1}}$ pin has not occurred

After servicing this interrupt, this flag should be cleared by writing a “1” to this bit.

INT2F

1 (set) – External interrupt on $\overline{\text{INT2}}$ pin has occurred

0 (clear) – External interrupt on $\overline{\text{INT2}}$ pin has not occurred

After servicing this interrupt, this flag should be cleared by writing a “1” to this bit.

5.5 Miscellaneous Control Register ($27)

Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
$27	INT2N	INT1N	IRQN	TIMI				WDOG	0000 ---0

IRQN

1 (set) – Edge triggering for $\overline{\text{IRQ}}$ only

0 (clear) – Level and edge triggering for $\overline{\text{IRQ}}$

INT1N

1 (set) – Edge triggering for $\overline{\text{INT1}}$ only

0 (clear) – Level and edge triggering for $\overline{\text{INT1}}$

INT2N

1 (set) – Edge triggering for $\overline{\text{INT2}}$ only

0 (clear) – Level and edge triggering for $\overline{\text{INT2}}$

5.6 Programmable Timer Interrupt

Three timer interrupt flags are found in the four most significant bits of the Timer Status register (TSR) at location $13. All three interrupts will vector to the same address at location $3FF6-$3FF7.

Each flag bit is defined as follows:

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Timer Status Register	$13	ICF	OCF	TOF						uuu- ----

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 ($19). Reset does not affect this bit.

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 ($17). Reset does not affect this bit.

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 ($15). Reset does not affect this bit.

All three timer interrupt flags have corresponding enable bits (ICIE, OCIE, and TOIE) found in the Timer Control Register (TCR) at location $12. 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 $3FF6 and $3FF7.

Refer to section 7 - Programmable Timer for detailed description.

5.7 RTC Interrupt

An RTC interrupt is enabled when either the RTCE, ALE or SECE bit in the RTC Control register ($1D) is set, provided the interrupt mask bit in the Condition Code register is cleared. When RTCE bit is set, real time clock will interrupt the CPU once a day. This will occur when the hours register in real time clock register changes from twenty-three to zero. When the SECE bit is set, the real time clock will interrupt CPU once a second. When ALE bit is set, RTC interrupt will occur when the value of hour alarm and hours are equal, and the value of minute alarm and minutes are equal. 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 mask further interrupts until the present one is serviced. The interrupt causes the program counter to vector to $3FF2 - $3FF3 in which is stored the service routine's starting address. Interrupt per day, per second or alarm interrupt can be distinguished by the RTCF, SECF and ALF flags. In order to reset the interrupt, users are responsible for clearing the appropriate flags when executing the interrupt routine.

Notice that ALF is set as long as the values of the both hour alarm and hour, minute alarm and minutes are qual. Therefore, users may found ALF is still set even after the MCU performs an ALF flag clearance. To avoid the same alarm interrupt from further interrupting the MCU, users can assign a different values to the alarm register and then ALF can be cleared as a result.

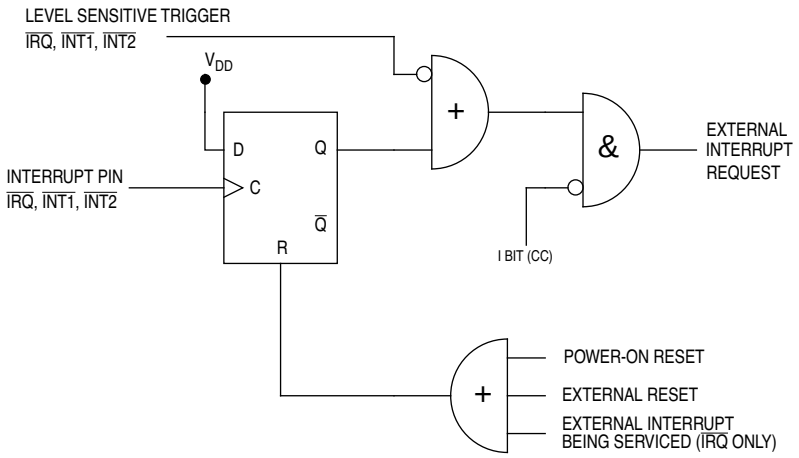
Refer to section 6.2 - Real Time Clock for detailed description.

5.8 SPI Interrupt

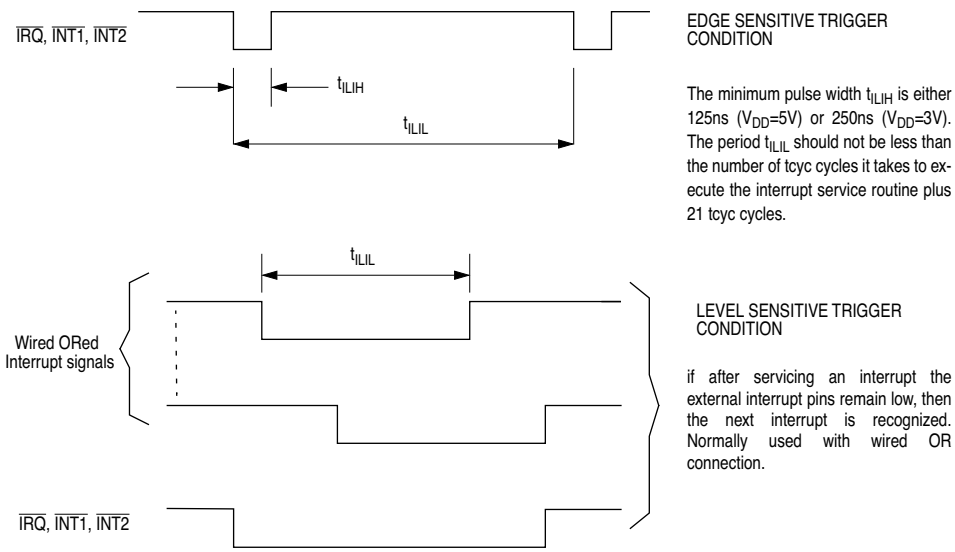
An interrupt in the serial peripheral interface (SPI) occurs when one of the interrupt flag in the serial peripheral status register ($2B) is set, provided the interrupt mask bit in the Condition Code

register is cleared and the enable bit in the serial peripheral control register ($2A) is enabled. 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 SPI interrupt causes the program counter to vector to memory location $3FF4 and $3FF5 which contains the starting address of the interrupt's service routine. Software in the serial peripheral interrupt service routine must determine the priority and the cause of the SPI interrupt by examining the interrupt flags located in the SPI status register. The general sequence of clearing an interrupt is a software sequence of accessing the status register while the flag is set, followed by a read or write of an associated register.

Refer to section 8 - Serial Peripheral Interface for detailed description.



(a) Interrupt Function Diagram



(b) Interrupt Mode Diagram

Figure 5-4 External Interrupt Circuit and Timing

6
CLOCKS

This section describes the Phase Lock Loop and Real Time Clock.

6.1 PHASE LOCK LOOP

System clock can be obtained from either the 32KHz oscillator or the PLL. During power-up or external reset, CPU system clock comes from the 32KHz clock. Setting CLKS of the PLL Status/Control register (bit 7 of address $24) selects PLL clock for the CPU. PLL clock frequency depends on FS1 & FS0; see Table 6-1. The Phase Lock Loop Indicator, PLLI (bit 0), is a read only bit which indicates an accurate clock is ready when set. However, owing to the finite delay in updating this PLLI, it may still be set after FS1 & FS0 are modified and the PLL clock frequency is not yet stabled. FS1 & FS0, and CLKS are cleared during power-up or external reset.

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
PLL Status and Control Register	$24	CLKS	FS1	FS0	STPOSC				PLLI	0000 ---u

CLKS - Clock Option

1 (set) – Use the PLL clock as the CPU clock source

0 (clear) – Use the 32KHz clock as the CPU clock source

Table 6-1 Phase Lock Loop Output Frequencies

FS1	FS0	PLL Output Clock Frequency	PO2 (Internal Bus Frequency)
0	0	0.328MHz	0.164MHz
0	1	2.622MHz	1.311MHz
1	0	0.656MHz	0.328MHz
1	1	5.244MHz	2.622MHz

STPOSC - Stop Oscillator

1 (set) – Oscillator is disabled in Stop mode

0 (clear) – Oscillator is enabled in Stop mode

PLLI

1 (set) – PLL clock is stable

0 (clear) – PLL clock is not yet stable

CLKS is the bit to select CPU clock either coming from the phase lock loop or from the 32KHz clock. During power-up or external reset, this bit is cleared to indicate that CPU clock is from the 32KHz clock. Setting the STPOSC bit will disable the oscillator in Stop mode. The following is an example of how to use the PLL to obtain an accurate CPU clock.

```
BSET   7,$24        Select PLL clock for CPU.
BSET   5,$24        Select 1.302 MHz as internal bus frequency.
JSR    DELAY        Wait a moment due to finite delay of PLLI bit
BRCLR  0,$24,*      Wait until PLL clock is stable.
JMP    SERVE        Clock is now stable, go to perform jobs that
                    require accurate frequency (such as Timer).
```

The PLL consists of an on-chip VCO, a phase comparator and a programmable divide-by-N counter. A filter is required to filter the phase comparator output to provide a DC signal to control the VCO frequency; see Figure 6-1.

The phase comparator compares the rising edge of an 1KHz reference signal derived from the 32KHz crystal clock to the rising edge of the VCO clock after being divided by the divide-by-N counter. When there is phase different between the two signals, the phase comparator output will adjust the DC level input to the VCO to change the VCO frequency; see Figure 6-2. The divide-by-N counter can be programmed to divide by different rates to obtain 4 different VCO frequencies of 288KHz, 576KHz, 2.604MHz, and 5.208MHz. Refer to Figure 12-3 and Figure 12-4 for external low-pass filter components.

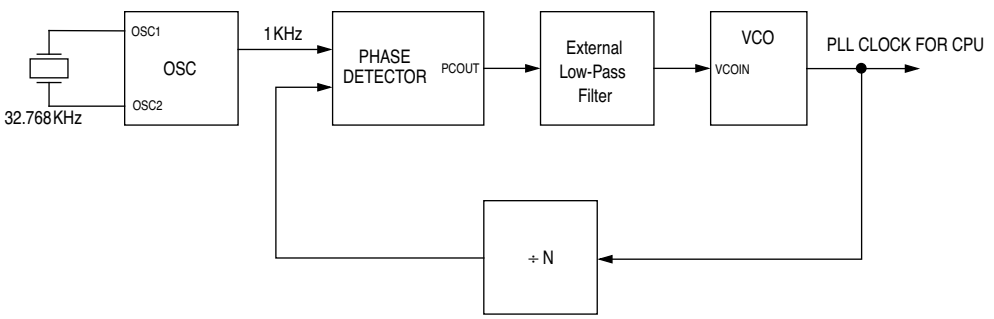


Figure 6-1 Phase Lock Loop Block Diagram

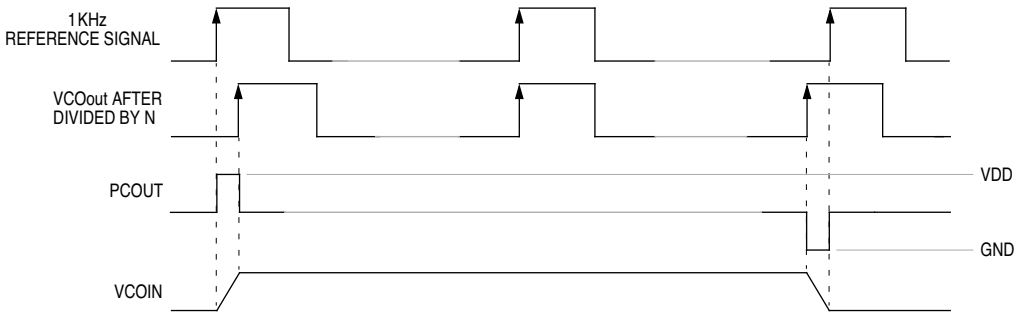


Figure 6-2 Typical Waveform for PLL

6.2 REAL TIME CLOCK

Continuous or non-continuous RTC operation can be selected by programming the STPOSC bit in the PLL Status & Control register (bit 4 of address $24). Real time clock consists of three binary counters which are divided down from the 32.768 KHz oscillator. There are three bits in the RTC status register (address $1C) and three bits in the RTC Control register (address $1D) where the operation of the real time clock is controlled.

6.2.1 RTC Time Registers

Three locations are reserved for real time clock operation; they are:

RTC Hours

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

RTC Minutes

Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
$10								

RTC Seconds

Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
$11								

The hours and minutes registers are in fact holding registers (See Figure 6-3). When setting the RTC time, the hours and minutes are written first, then writing to the seconds register will cause the hours and minutes in their holding registers to be latched into the RTC hardware.

When reading the RTC time, the seconds register is first accessed; this will cause the hardware to latch the hours and minutes into their respective holding registers. Now, reading the hours and minutes registers will give the time for the moment when the seconds register was read.

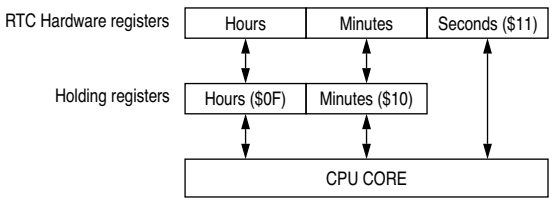


Figure 6-3 RTC Holding Registers

6.2.2 RTC Alarm Registers

There are three locations associated with the alarm registers. They are:

RTC Hour Alarm

Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on Reset
$0C									0000 0000

RTC Minute Alarm

Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on Reset
$0D									0000 0000

RTC Second Alarm

Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on Reset
$0E									0000 0000

The ALF bit in the RTC Status register is set when the contents of the RTC Alarm registers matches the contents of the RTC Time registers.

6.2.3 RTC Status Register

There are three interrupts associated with the RTC, their flags are in the RTC Status register ($1C), and their respective enable bits are the RTC Control register ($1D).

Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on Reset
$1C						SECF	ALF	RTCF	---- -000

RTCF

1 (set) – Indicates once a day interrupt has occurred. After serving this interrupt, it is the user's responsibility to clear this bit, otherwise the CPU will keep on serving this once a day interrupt when a new RTC interrupt occurs (even there is no once a day interrupt occurs).

0 (clear) – Indicates once a day interrupt has not occurred.

ALF

1 (set) – Indicates alarm interrupt has occurred. After serving this interrupt, it is the user's responsibility to clear this bit, otherwise the CPU will keep on serving this alarm interrupt when a new RTC interrupt occurs (even there is no alarm interrupt occurs).

0 (clear) – Indicates alarm interrupt has not occurred.

SECF

1 (set) – Indicates once a second interrupt has occurred. After serving this interrupt, it is responsibility to clear this bit, otherwise the CPU will keep on serving this once a second interrupt when a new RTC interrupt occurs (even there is no once a second interrupt occurs).

0 (clear) – Indicates once a second interrupt has not occurred.

6.2.4 RTC Control Register

Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on Reset
$1D						SECE	ALE	RTCE	---- -000

RTCE

1 (set) – Real time clock once a day interrupt enable

0 (clear) – Real time clock once a day interrupt disable

ALE

1 (set) – Alarm interrupt enable

0 (clear) – Alarm interrupt disable

SECE

1 (set) – Real time clock once a second interrupt enable

0 (clear) – Real time clock once a second interrupt disable

It should be noted that the flags in the interrupt status register will be set if the corresponding event is detected, irrespective of the setting of the interrupt enable bits.

Following is an example showing how to use the RTC interrupt:

```
*Main program
       BSET   0,$1D          Enable RTC (once a day) interrupt.
       BSET   1,$1D          Enable RTC (alarm) interrupt.
       BSET   2,$1D          Enable RTC (once a second) interrupt.
       STOP                  MCU execute STOP instruction for power
                             conservation.
*Real time clock interrupt service routine
       BRSET  0,$1C,ODAY     This bit is set indicating this is an once a
                             day RTC interrupt.
       BRSET  1,$1C,ALINT    This bit is set indicating this is a RTC
                             interrupt caused by a match of alarm registers
                             and real time clock registers.
       BRSET  2,$1C,OSEC     This bit is set indicating this is an once a
                             second RTC interrupt.
ODAY   BSET   0,$1C          Clear this bit so that this once a day
                             interrupt will not be recognized as a new one
                             on next RTC interrupt.
       JSR    OADAY          Once a day interrupt service routine.
       BRSET  1,$1C,ALINT    This bit is set indicating alarm interrupt
                             also occurs at the same time.
       BRSET  2,$1C,OSEC     This bit is set indicating once a second RTC
                             interrupt also occurs at the same time.
RTCR   RTI
ALINT  BSET   1,$1C          Clear this bit so that this alarm interrupt
                             will not be recognized as a new one on next
                             RTC interrupt.
       JSR    ALARM          Alarm service interrupt routine.
       BRSET  2,$1C,OSEC     This bit is set indicating once a second
                             interrupt also occurs at the same time.
       BRA    RTCR           Return from interrupt.
```

```
OSEC    BSET    2,$1C       Clear this bit so that this once a second
                            interrupt will not be recognized as a new one
                            on next RTC interrupt.
        JSR     OASEC       Once a second interrupt service routine.
        BRA     RTCR        Return from interrupt.
```

THIS PAGE LEFT BLANK INTENTIONALLY

7
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 7-1 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) $12
- Timer Status Register (TSR) $13
- Input Capture Register High byte - $14, Low byte - $15
- Output Compare Register High byte - $16 Low byte - $17
- Counter Register High byte - $18, Low byte - $19
- Alternate Counter Register High byte - $1A, Low byte - $1B

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

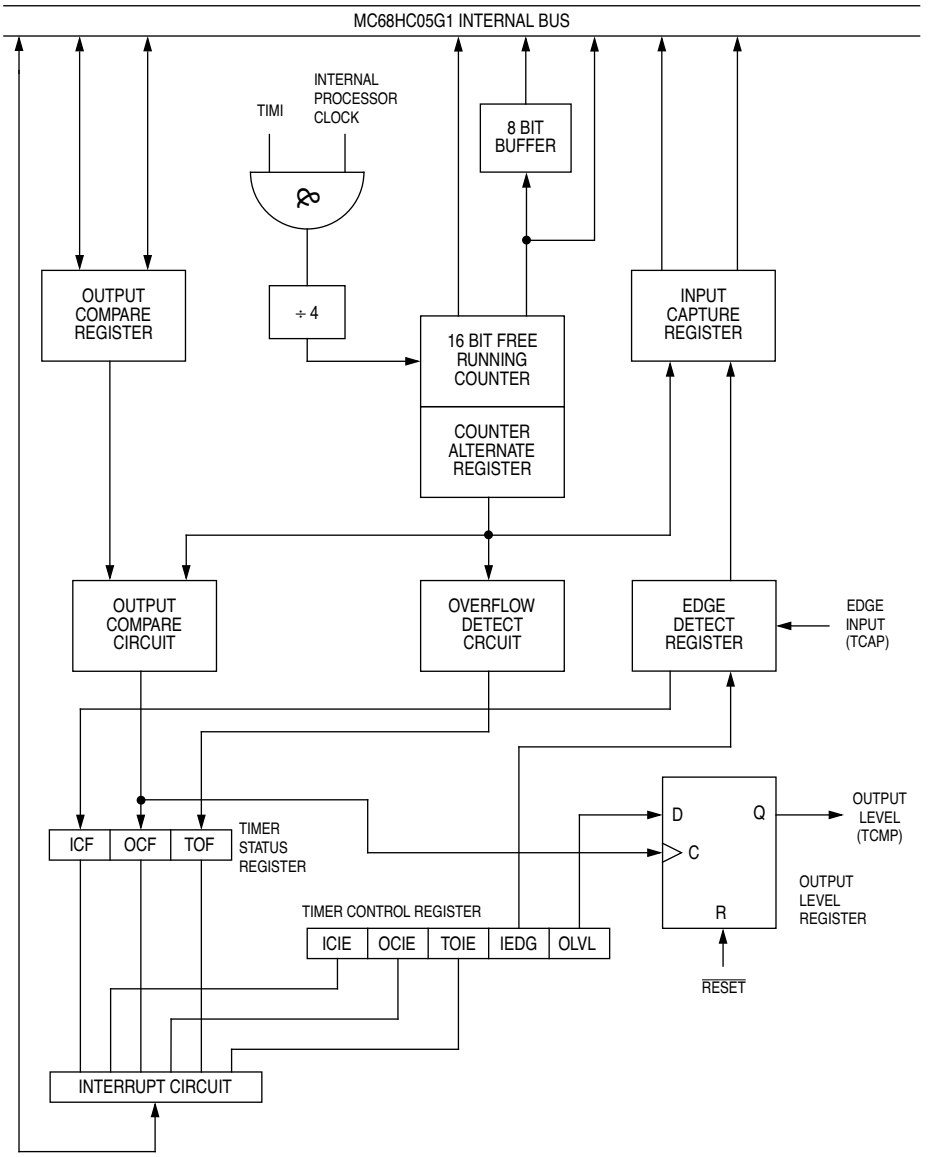


Figure 7-1 Programmable Timer Block Diagram

7.1 Counter

– Counter Register location High byte - $18, Low byte - $19
– Alternate Counter Register High byte - $1A Low byte - $1B

The key element in the programmable timer is a 16-bit, free-running counter or counter register, preceded by a prescaler that divides the internal processor clock by four. The prescaler gives the timer a resolution of 3.472µs if the internal bus clock is 1.152MHz. The counter is incremented during the low portion of the internal bus clock. Software can read the counter at any time without affecting its value.

The double-byte, free-running counter can be read from either of two locations, $18 & $19 (counter register) or $1A & $1B (counter alternate register). Reading only the least significant byte (LSB) of the free-running counter ($19 or $1B) receives the count value at the time of the read. If the most significant byte (MSB) ($18 or $1A) is read first, the LSB ($19 or $1B) 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 ($19 or $1B) 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 internal bus clock cycles. TOF is set when the counter overflows (from $FFFF to $0000); this will cause an interrupt if TOIE (bit 5 of Timer Control register) is set.

In some timing control applications it may be desirable to reset the counter under software control. When the low byte of the counter ($19 or $1B) is written to, the counter is set to its reset value of $FFFC. The divide-by-4 prescaler is also reset and the counter resumes normal counting operation. All of the flags and enable bits remain unaltered by this operation. If access has previously been made to the high byte of the free-running counter ($18 or $1A), then the reset counter operation terminates the access sequence.

7.2 Output Compare Registers

– Output Compare Register High byte - $16 Low byte - $17

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 four internal bus clock cycles.

After a processor write cycle to the output compare register containing the MSB ($16), the output compare function is inhibited until the LSB ($17) is also written. The user must write both bytes (locations) if the MSB is written first. A write made only to the LSB ($17) will not inhibit the compare function. The 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.

7.3 Input Capture Registers

– Input Capture Register High byte - $14, Low byte - $15

'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 one count of the free-running counter, which is four internal bus clock cycles.

The free-running counter contents are transferred to the input capture register on each 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 ($14), the counter transfer is inhibited until the LSB ($15) 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 ($15) does not inhibit the free-running counter transfer since they occur on opposite edges of the internal bus clock.

7.4 Timer Control Register (TCR)

Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on Reset
$12	ICIE	OCIE	TOIE				IEDG	OLVL	000- --u0

The TCR is a read/write register containing five control bits. Three bits control interrupts associated with each of 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.

The programmable timer may be disabled/enabled by writing to bit 4 in the Miscellaneous Control register.

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Miscellaneous Control Register	$27	INT2N	INT1N	IRQN	TIMI				WDOG	0000 ---0

TIMI

1 (set) – Timer inhibit

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

7.5 Timer Status Register (TSR)

Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on Reset
$13	ICF	OCF	TOF						uuu- ----

The Timer Status register ($13) 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 ($15)

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 ($17).

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 $12) is set. TOF is cleared by reading the TSR and the counter low register ($19).

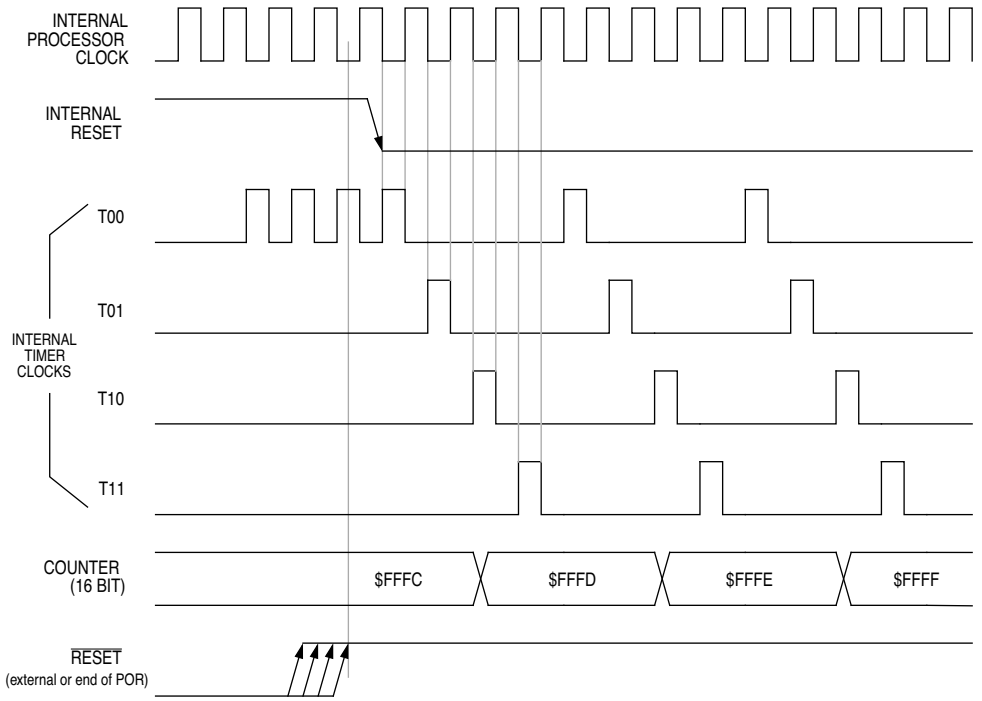
When using the timer overflow function and reading the free-running counter at random times to measure an elapsed time, a problem may occur whereby the timer overflow flag is unintentionally cleared if:

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.

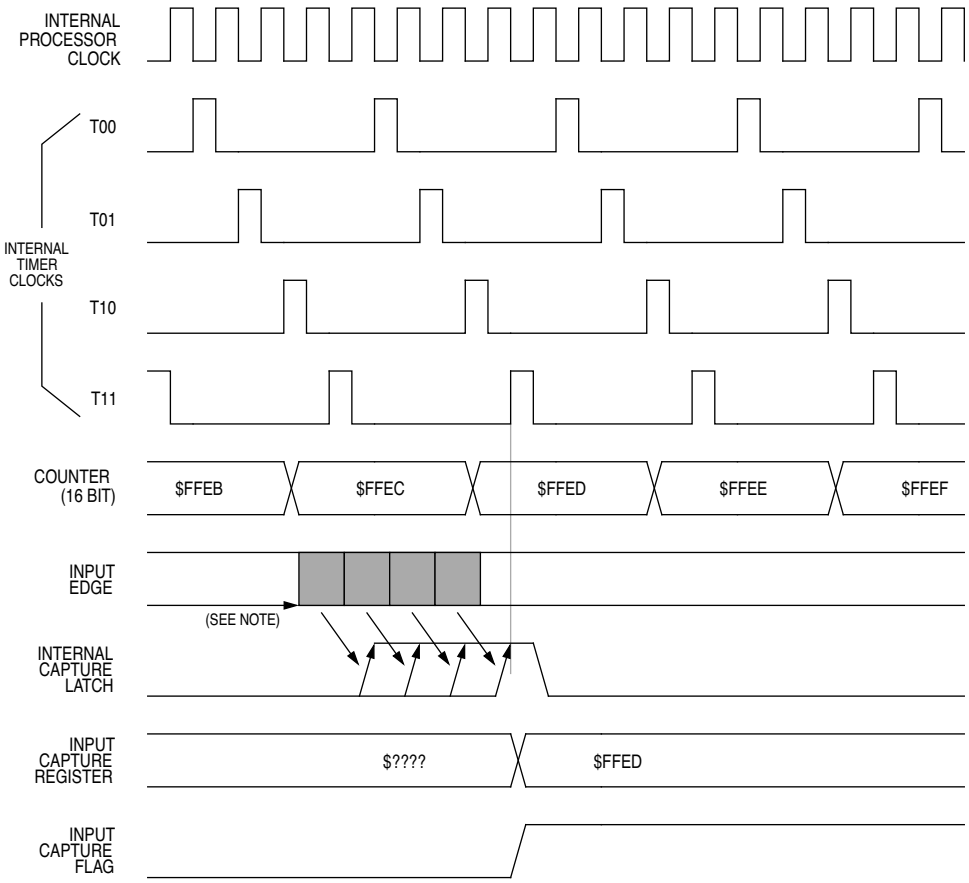
7.6 Programmable Timer Timing Diagrams

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.



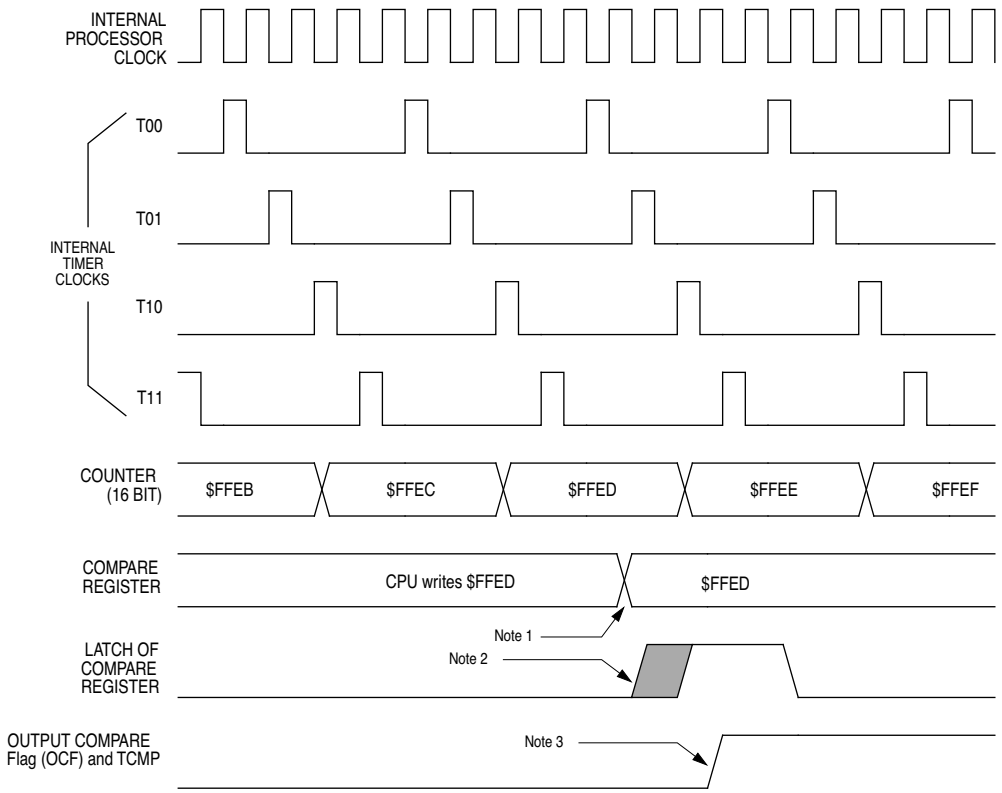
Note: RESET affects only the counter register and timer control register.

Figure 7-2 Timer State Timing Diagram for Reset



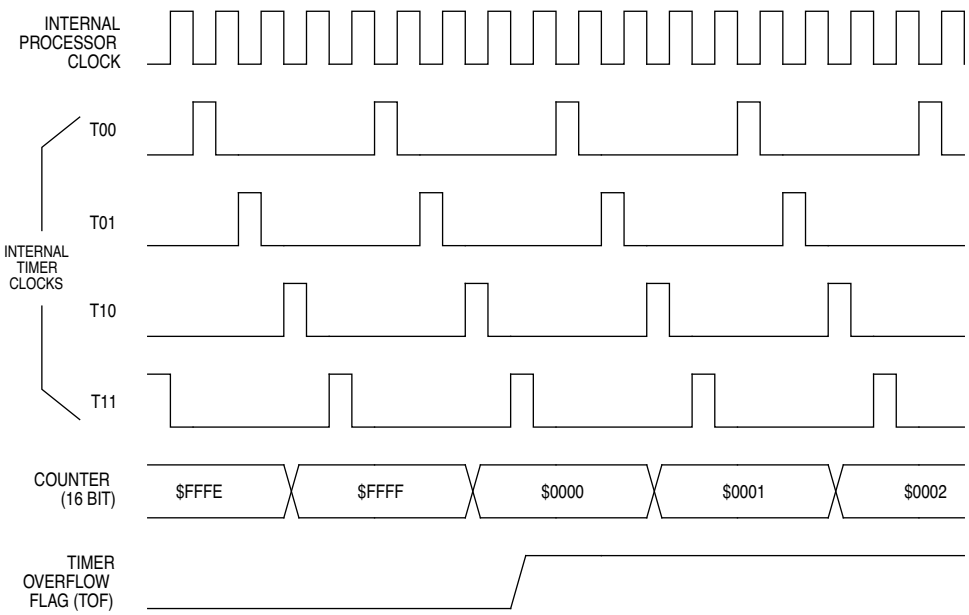
Note: If the input edge occurs in the shaded area from one timer state T10 to the other timer state T10 the input capture flag is set during the next state T11.

Figure 7-3 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 either on of the compare registers and the actual compare.
2. Internal compare takes place during timer state T01.
3. OCF is set at timer state T11 which follows the comparison match ($FFED in this example).

Figure 7-4 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 7-5 Timer State Diagram for Timer Overflow

THIS PAGE LEFT BLANK INTENTIONALLY

8
SERIAL PERIPHERAL INTERFACE

The serial peripheral interface (SPI) is an interface built into the MC68HC05G1 microcontroller which allows several SPI microcontrollers, or SPI-type peripherals to be interconnected within a single “black box” or on the same printed circuit board. In a serial peripheral interface, separate wires (signals) are required for data and clock. In the SPI format, the clock is not included in the data stream and must be furnished as a separate signal. An SPI system may be configured as one containing one master MCU and several slave MCUs, or as a system in which an MCU is capable of being either a master or slave.

Figure 8-1 illustrates two different system configurations. Figure 8-1(a) represents a system of five different microcontrollers in which there is one master and four slaves (0,1,2,3). In this system four basic lines (signals) are required for MOSI (master out slave in), MISO (master in slave out), SCK (serial clock), and SS (slave select) lines. Figure 8-1(b) represents a system of five microcomputers in which three can be either master of slave and two are slave only.

8.1 SPI Features

- Full Duplex, Three-Wire Synchronous Transfers
- Master or Slave Operation
- 1.288 MHz (Max.) Master Bit Frequency
- 2.4576 MHz (Max.) Slave Bit Frequency
- Four programmable Master Bit Rates
- Programmable Clock Polarity and Phase
- End of Transmission Interrupt Flag
- Data Collision Flag Protection
- Master-Master Mode Fault Protection Capability

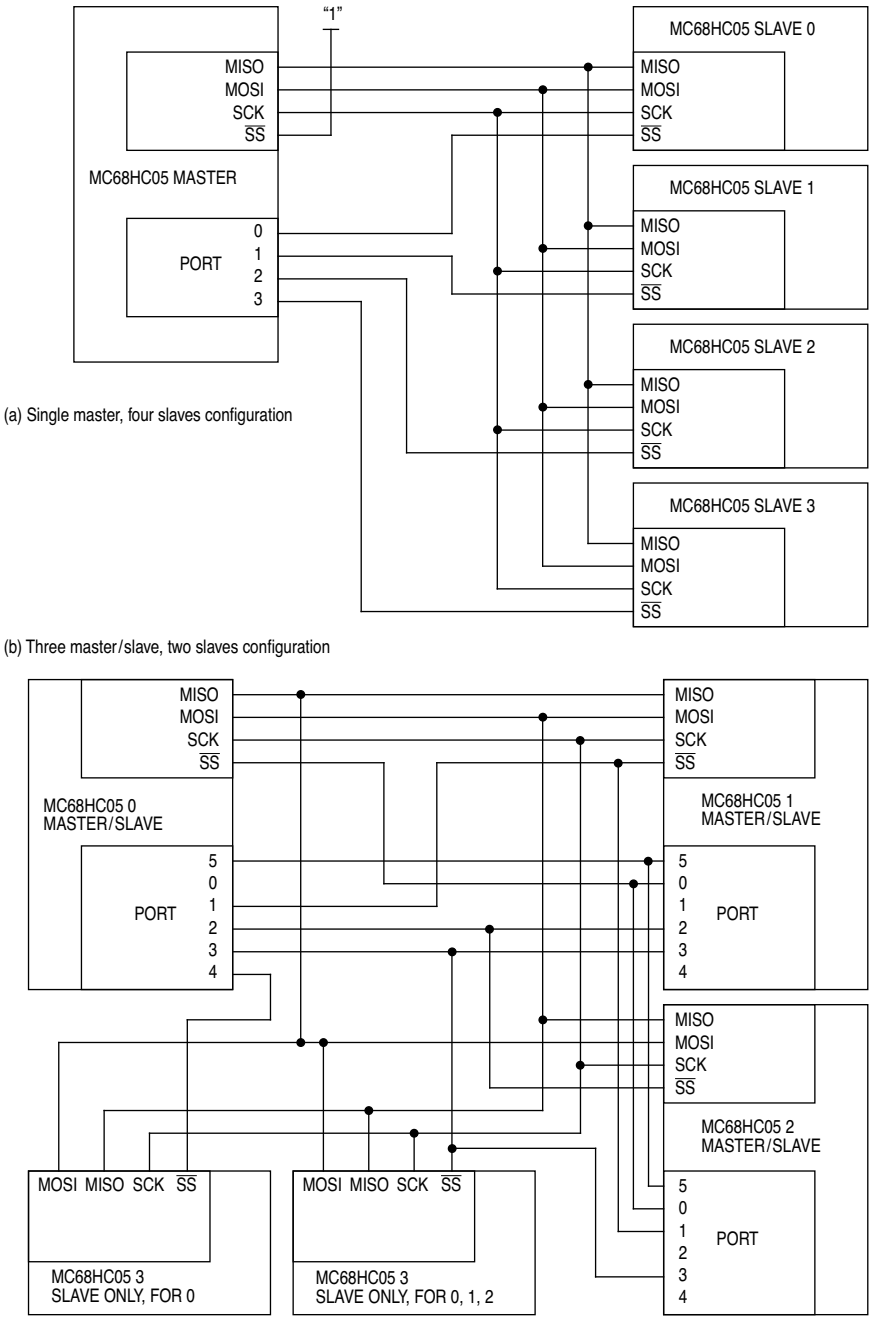


Figure 8-1 SPI Master-Slave System Configurations

8.2 SPI Signals Description

The four basic SPI signals (MOSI, MISO, SCK, and $\overline{SS}$) are described in the following paragraphs. Each signal is described for both the master and slave mode.

8.2.1 Master Out Slave In (MOSI)

The MOSI pin is configured as an output in a master (mode) device and as an input in a slave (mode) device. Data is transferred serially from a master to a slave on this line, most significant bit first. The timing diagram of Figure 8-2 shows the relationship between data and serial clock (SCK). As shown in Figure 8-2, four possible timing relationships may be chosen by using control bits CPOL and CPHA. The master device always allows data to be applied on the MOSI line a half-cycle before the serial clock edge (SCK) in order for the slave device to latch the data.

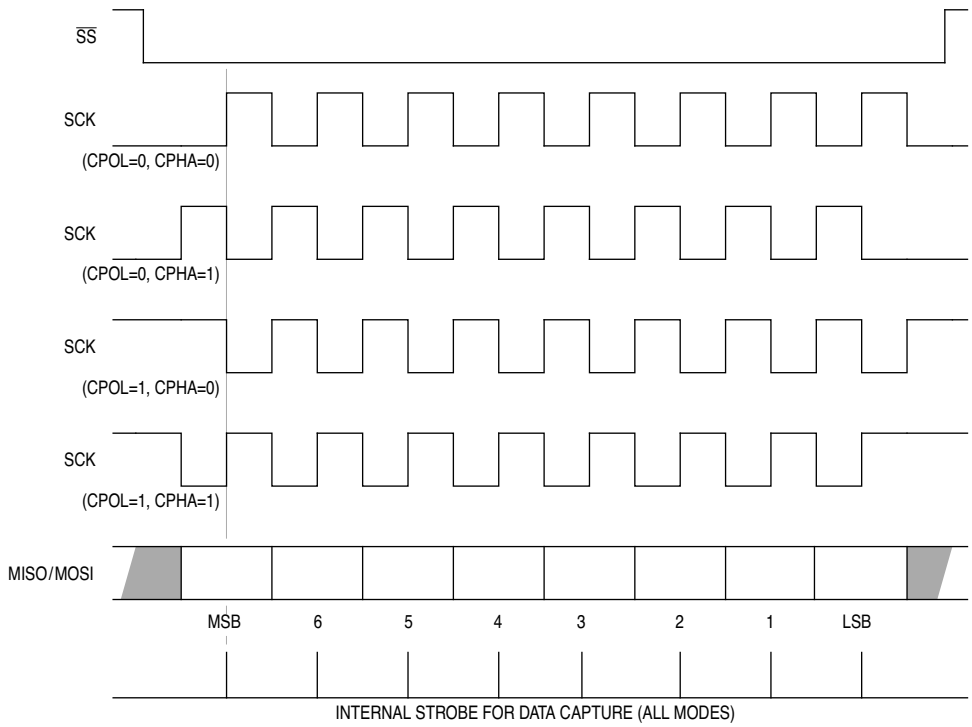


Figure 8-2 SPI Data/Clock Timing Diagram

Note: Both the slave device(s) and a master device must be programmed to similar timing modes for proper data transfer.

When the master device transmits data to a slave via a MOSI line, the slave device responds by sending data to the master device via the MISO line. This implies full duplex transmission with both data out and data in synchronized with the same clock signal (one which is provided by the master device). Thus the byte transmitted is replaced by the byte received and eliminates the need for separate transmit-empty and receiver-full status bits. A single status bit (SPIF) in the serial peripheral status register (SPSR, location $2B) is used to signify that the I/O operation is complete.

Configuration of the MOSI pin is a function of the MSTR bit in the serial peripheral control register (SPCR, location $2A). When a device is operating as a master, the MOSI pin is an output because the program in firmware set the MSTR bit as a logic one.

8.2.2 Master In Slave Out (MISO)

The MISO pin is configured as an input in a master (mode) device and as an output pin in a slave (mode) device. Data is transferred serially from a slave to a master on this line, most significant bit first. The MISO pin of a slave is placed in a high-impedance state if it is not selected by a master; i.e., its $\overline{SS}$ pin is a logic one. The timing diagram in Figure 8-2 shows the relationship between data and serial clock (SCK). As shown in Figure 8-2, four possible timing relationships may be chosen by using control bits CPOL and CPHA. The master device always allows data to be applied on the MISO line a half-cycle before the serial clock edge (SCK) in order for the slave device to latch the data.

Note: Both the slave device(s) and a master device must be programmed to similar timing modes for proper data transfer.

When a master device transmits data to a slave device via the MOSI line, the slave device responds by sending data to the master device via the MISO line. This implies full duplex transmission with both data out and data in synchronized with the same clock signal (one which is provided by the master device). Thus, the byte transmitted is replaced by the byte received and eliminates the need for separate transmit-empty and receiver-full status bits. A single status bit (SPIF) in the serial peripheral status register (SPSR, location $2B) is used to signify that the I/O operation is complete.

In the master device, the MSTR control bit in the serial peripheral control register (SPCR, location $2A) is set to a logic one (by the program) to allow the master device to receive data on its MISO pin. In the slave device, its MISO pin is enabled by the logic level of the $\overline{SS}$ pin; i.e., if $\overline{SS}$=1 then the MISO pin is placed in a high impedance state, whereas, if $\overline{SS}$=0 the MISO pin is an output for the slave device.

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

The slave select ($\overline{SS}$) pin is a fixed input which receives an active low signal that is generated by the master device to enable slave device(s) to accept data. To ensure that data will be accepted by a slave device, the $\overline{SS}$ signal line must be a logic low prior to occurrence of serial SCK and must remain low until after the last (eighth) SCK cycle. Figure 8-2 illustrates the relationship between SCK and the data for two different level combinations of CPHA, when $\overline{SS}$ is pulled low. These are: 1) with CPHA = 1 or 0, the first bit or data is applied to the MISO line for transfer, and 2) when CPHA = 0 the slave device is prevented from writing to its data register. Refer to the WCOL status flag in the serial peripheral status register (bit 6, location $2B) description for further information on the effects that the $\overline{SS}$ input and CPHA control bit have on the I/O register. A high level $\overline{SS}$ signal forces the MISO line to the impedance state. Also, SCK and the MOSI line are ignored by a slave device when its $\overline{SS}$ signal is high.

When the device is a master, it constantly monitors its $\overline{SS}$ signal input for a logic low. the master device will become a slave device any time its $\overline{SS}$ signal is detected low. This ensures that there is only one master controlling the $\overline{SS}$ line for a particular system. When the $\overline{SS}$ line is detected low, it clears the MSTR control bit in the SPSR (bit 4, location $2A). Also, control bit SPE in the SPCR (bit 6, location $2A) is cleared and causes the serial peripheral interface to be disabled. The MODF flag bit in the serial peripheral status register (location $2B) is also set to indicate to the master device that another device is attempting to become a master. Two devices attempting to be outputs are normally the result of a software error; however, a system could be configured which would contain a default master which would automatically “take-over” and restart the system.

8.2.4 Serial Clock (SCK)

The serial clock is used to synchronize the movement of data both in and out the device through its MOSI and MISO pins. The master and slave devices are capable of exchanging a data byte of information during a sequence of eight clock pulses. Since SCK is generated by the master device, the SCK line becomes an input in all slave devices and synchronizes slave data transfer. The type of clock and its relationship to data are controlled by the CPOL and CPHA bits in the SPCR (location $2A). Refer to Figure 8-2 for timing.

The master device generates the SCK through a circuit driven by the internal processor clock. Two bits (SPR0 and SPR1) in the SPCR (location $2A) of the master device select the clock rate. The master device uses the SCK to latch incoming slave device data on the MISO line and shifts out data to the slave device on the MOSI line. Both master and slave devices must be operated in the same timing mode as controlled by the CPOL and CPHA bit in the SPCR. In the slave device, SPR0, SPR1 have no effect on the operation of the serial peripheral interface. Timing is shown in Figure 8-2.

8.3 Functional Description

A block diagram of the serial peripheral interface is shown in Figure 8-3. In a master configuration, the master start logic receives an input from the CPU (in form of a write to the SPI rate generator) and originates the serial clock (SCK) based on the internal processor 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 internal bus) during a write cycle and then shifted out serially to the MOSI pin for application to the slave device(s). During a read cycle, data is applied serially from a slave device via the MISO 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 during a CPU read cycle.

In a slave configuration, the slave start logic receives a logic low (from a master device) at the $\overline{SS}$ pin and serial clock input (from the same master device) at the SCK pin. Thus, the slave is synchronized with the master. Data from the master is received serially at the slave MOSI pin 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 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 MISO pin for application to the master device.

Figure 8-4 illustrates the MOSI, MISO, and SCK master-slave interconnections. Note that in Figure 8-4 the master $\overline{SS}$ pin is tied to a logic high and the slave $\overline{SS}$ pin is a logic low. Figure 8-1 provides a larger system connection for these same pins. Note that in Figure 8-1, all $\overline{SS}$ pins are connected to a port pin of a master/slave device. In this case any of the devices can be a slave.

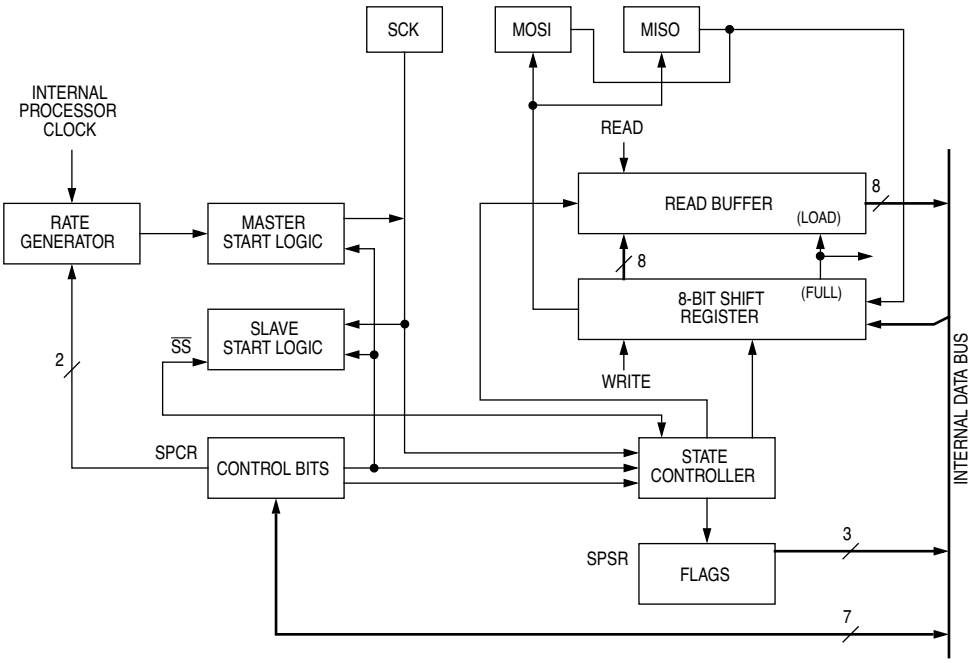
8.4 SPI Registers

These are three registers in the serial parallel interface which provide control, status, and data storage functions. These registers are the serial peripheral control register (SPCR, location $2A), serial peripheral status register (SPSR, location $2B), and serial peripheral data I/O register (SPDR, location $2C) are described in the following paragraphs.

8.4.1 Serial Peripheral Control Register (SPCR)

Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on Reset
$2A	SPIE	SPE		MSTR	CPOL	CPHA	SPR1	SPR0	00-0 uuuu

The serial peripheral control register bits are defines as follows:



Note: $\overline{SS}$, SCK, MOSI, and MISO are external pins; where

$\overline{SS}$ - provides a logic low to select a slave device for a transfer with a master device.

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

MOSI - provides serial output to slave unit(s) when device is configured as a master. Receives serial input from master when device is configured as a slave unit.

MISO - receives serial input from slave unit(s) when device is configured as a master. Provides serial output to master when device is configured as a slave unit.

Figure 8-3 Serial Peripheral Interface Block Diagram

8.4.1.1 SPIE - Serial Peripheral Interrupt Enable

1 (set) – Serial Peripheral Interrupt Enabled

0 (clear) – Serial Peripheral Interrupt Disabled

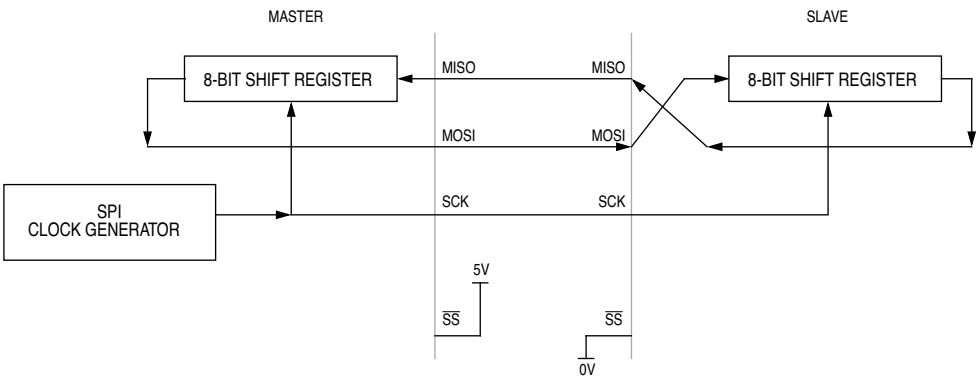


Figure 8-4 SPI Master-Slave Interconnection

8.4.1.2 SPE - Serial Peripheral Output Enable

1 (set) – Port D pins 2, 3, 4, 5 configured as SPI pins

0 (clear) – Port D pins 2, 3, 4, 5 configured as standard I/O pins

When the serial peripheral output enable control bit is set, all output drive is applied to the external pins and the system is enabled. When the SPE bit is set, it enables the SPI system by connecting it to the external pins thus allowing it to interface with the external SPI bus. The pins that are defined as output depend on which mode (master or slave) the device is in. Because the SPI bit is cleared by reset, the SPI system is not connected to the external pins upon reset.

8.4.1.3 MSTR - Master

1 (set) – SPI configured as master

0 (clear) – SPI configured as slave

The master bit determines whether the device is a master or a slave. If the MSTR bit is a logic zero it indicates a slave device and a logic one denotes a master device. If the master mode is selected, the function of the SCK pin changes from an input to an output and the function of the MSIO and MOSI pins are reversed. This allows the user to wire device pins MISO to MISO, and MOSI to MOSI, and SCK to SCK without incident. The MSTR bit is cleared by reset; therefore, the device is always placed in the slave mode during reset.

8.4.1.4 CPOL - Clock Polarity

1 (set) – SCK pin high during master idle

0 (clear) – SCK pin low during master idle

The clock polarity bit controls the normal or steady state logic of the clock when data is not being transferred. The CPOL bit affects both the master and slave modes. It must be used in conjunction with the clock phase control bit (CPHA) to produce the desired clock-data relationship between a master and a slave device. When the CPOL bit is a logic zero, it produces a steady state low value at the SCK pin of the master device. If the CPOL bit is a logic one, a high value is produced at the SCK pin of the master device when data is not being transferred. The CPOL bit is not affected by reset. Refer to Figure 8-2.

8.4.1.5 CPHA - Clock Phase

1 (set) – 2nd clock edge transition

0 (clear) – 1st clock edge transition

The clock phase bit controls the relationship between the data on the MISO and MOSI pins and the clock produced or received at the SCK pin. This control has effect in both the master and slave modes. It must be used in conjunction with the clock polarity control bit (CPOL) to produce the wanted clock-data relation. The CPHA bit in general selects the clock edge which captures data and allows it to change states. It has its greatest impact on the first bit transmitted (MSB) in that it does or does not allow a clock transition before the first data capture edge. The CPHA bit is not affected by reset. Refer to Figure 8-2.

8.4.1.6 SPR1, SPR0 - Serial Peripheral Rate

These two serial peripheral rate bits select one of four baud rates to be used as SCK if the device is a master; however, they have no effect in slave mode. The slave device is capable of shifting data in and out at a maximum rate which is equal to the CPU clock. A rate table is given below for the generation of the SCK from the master. The SPR1 and SPR0 bits are not affected by reset.

SPR1	SPR0	INTERNAL PROCESSOR CLOCK DIVIDE BY
0	0	2
0	1	4
1	0	16
1	1	32

8.4.2 Serial Peripheral Status Register (SPSR)

Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on Reset
$2B	SPIF	WCOL		MODF					00-0 ----

This is a read-only register. The status flags which generate a serial peripheral interface (SPI) interrupt may be blocked by the SPIE control bit in the serial peripheral control register. The WCOL bit does not cause an interrupt. The serial peripheral status register bits are defined as follows:

8.4.2.1 SPIF - Serial Peripheral Data Transfer Complete 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 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. During an overrun condition, when the master device has sent several bytes of data and the slave device has not responded to the first SPIF, only the first byte sent is contained in the receiver buffer and all other bytes are lost.

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 (using a similar sequence) during a second transmission; however, it must be cleared before the second SPIF in order to prevent an overrun condition. The SPIF bit is cleared by reset.

8.4.2.2 WCOL - Write Collision Status

The function of the write collision status bit is to notify the user that an attempt was made to write to the serial peripheral data register while a data transfer was taking place with an external device. The transfer continues uninterrupted; therefore, a write will be unsuccessful. A “read collision” will never occur since the received data byte is placed in a buffer in which access is always synchronous with the MCU operation. If a “write collision” occurs, WCOL is set but no SPI interrupt is generated. The WCOL is a status flag only.

Clearing the WCOL bit is accomplished by a software sequence of accessing the serial peripheral status register while WCOL is set, followed by 1), a read if the serial peripheral data register prior to the SPIF bit being set, or 2) a read or write of the serial peripheral data register after the SPIF bit is set. A write to the serial peripheral data register (SPDR) prior to the SPIF bit being set, will result in generation of another WCOL status flag. Both the SPIF and WCOL bits will be cleared in

the same sequence. If a second transfer has started while trying to clear (the previously set) SPIF and WCOL bits with a clearing sequence containing a write to the serial peripheral data register, only the SPIF bit will be cleared.

A collision of a write to the serial peripheral data register while an external data transfer is taking place can occur in both the master mode and the slave mode, although with proper programming the master device should have sufficient information to prevent this collision.

Collision in the master device is defined as a write of the serial peripheral data register while the internal rate clock (SCK) is in the process of transfer. The signal in the SS pin in always high on the master device.

A collision in a slave device is defined in two separate modes. One problem arises in a slave device when the CPHA control bit is a logic zero. When CPHA is a logic zero, data is latched with the occurrence of the first clock transition. The slave device does not have any way of knowing when that transition will occur; therefore, the slave device collision occurs when it attempt to write the serial peripheral data register after its $\overline{SS}$ pin has been pulled low. The $\overline{SS}$ pin of the slave device freezes the data in its serial peripheral data register and does not allow it to be altered if the CPHA bit is logic zero. The master device must raise the $\overline{SS}$ pin of the slave device high between each byte it transfer to the slave device.

The second collision mode is defined for the state of the CPHA control bit being a logic one. With the CPHA bit set, the slave device will be receiving a clock (SCK) edge prior to the latch of the first data transfer. This first clock edge will freeze the data in the slave device I/O register and allow the most significant bit onto the external MISO pin of the slave device. The $\overline{SS}$ pin low state enables the slave device but the drive onto the MOSI pin does not take place until the first data transfer clock edge. The WCOL bit will only be set if the I/O register is accessed while a transfer is taking place. By definition of the second collision mode, a master device might hold a slave device $\overline{SS}$ pin low during a transfer of several bytes of data without a problem.

A special case of write collision occurs in the slave device. This happens when the master device starts a transfer sequence (an edge of SCK for CPHA=1; or an active $\overline{SS}$ transition for CPHA=0) at the same time the slave device CPU is writing to its serial peripheral interface data register. In this case it is assumed that the data byte written (in the slave device serial peripheral interface) is lost and the contents of the slave device read buffer becomes the byte that is transferred. Because the master device receives back the last byte transmitted, the master device can detect that a fatal write collision occurred.

Since the slave device is operating asynchronously with the master device, the WCOL bit may be used as an indicator of a collision occurrence. This helps to alleviate the user from a strict real-time programming effort. The WCOL bit is cleared by reset.

8.4.2.3 MODF - Mode Fault

The function of the mode fault flag bit is defined for the master mode (device). If the device is a slave device the MODF bit will be prevented from toggling from a logic zero to a logic one; however, this does not prevent the device from being in the slave mode with MODF bit set. The

MODF bit is normally a logic zero and is set only when the master device has its $\overline{SS}$ pin pulled low. Toggling the MODF bit to a logic one affects the internal serial peripheral interface (SPI) system in the following ways:

1) MODF is set and SPI interrupt is generated if SPIE=1
2) The SPE bit is forced to a logic zero. This blocks all output drive from the device, disables the SPI system.
3) The MSTR bit is forced to a logic zero, thus forcing the device into the slave mode.

Clearing the MODF is accomplished by a software sequence of accessing the serial peripheral status register while MODF is set followed by a write to the serial peripheral control register. Control bits SPE and MSTR may restored to their original set state during this clearing sequence or after the MODF bit has been cleared. Hardware does not allow the user to set the SPE and MSTR bit while MODF is a logic one unless it is during the proper clearing sequence. The MODF flag bit indicates that there might have been a multi-master conflict for system control and allows a proper exit from system operation to a reset or default system state. The MODF is cleared by reset.

8.4.3 Serial Peripheral Data I/O Register (SPDR)

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

The serial peripheral data I/O register is used to transmit and receive data on the serial bus. 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 this data I/O register will not initiate a transmission. At the completion of transmitting a byte of data, the SPIF 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 of the slave device; all others are lost. The user may read the 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 WCOL and SPIF status bits to understand the limits on using the serial peripheral data I/O register.

8.5 SPI during Wait Mode

When the MCU enters the wait mode, the CPU clock is halted. All CPU action is suspended; however, the SPI system remain active. In fact an interrupt from the SPI (in addition to a logic low on the $\overline{IRQ}$, $\overline{INT1}$, $\overline{INT2}$, RTC, or a logic low on the $\overline{RESET}$ pin or a power on reset) causes the processor to exit the wait mode.

8.6 SPI during Stop Mode

When the processor executes the STOP instruction, the internal oscillator is turned off. This halts all internal CPU processing, including the operation of the serial peripheral interface. The only way for the MCU to "wake-up¨ from the stop mode is by receipt of an external interrupt (logic low on $\overline{IRQ}$, $\overline{INT1}$, $\overline{INT2}$), RTC, or the detection of a reset (logic low on $\overline{RESET}$ pin or a power-on reset).

When the MCU enters the stop mode, the baud rate generator which drives the SPI shuts down. This essentially stops all master mode SPI operation; 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 (provided it is an exit resulting from a logic low on the $\overline{IRQ}$, $\overline{INT1}$, $\overline{INT2}$ pin, interrupt from keyboard or RTC or by the detection of a reset of logic low on reset pin or a power on reset). If the stop mode is exited by a reset, then 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 and clock information in addition to transmitting its own data back to a master device.

Since the MC68HC05 is the bus master, it internally controls the function of its MOSI and MISO lines; thus writing data to the slave devices on the MOSI and reading data from the slave devices on the MISO lines. The master device selects the individual slave devices by using four pins of a parallel port to control the four $\overline{SS}$ pins of the slave devices. A slave device is selected when the master device pulls its $\overline{SS}$ pin low. The $\overline{SS}$ pins are pulled high during reset since the master device ports will be forced to be inputs at that time, thus disabling the slave devices. Note that the master can enable all slaves when writing to them, but can only read from one slave at a time. This is to prevent bus contention on the MISO line.

Example: in a one master, three slaves system, the master writes to the three slaves' display driver to clear a display with a single I/O operation. To ensure that proper data transmission between the master device and a slave device, the master device may have the slave device responding with a data byte previously sent by the master (this data byte could be inverted or at least be a byte that is different from the last one sent by the master device). The master device will always receive the byte back from the slave device if all MISO and MOSI lines are connected and the slave has not written to its data I/O register. Other transmission protocols may be defined using ports for handshake lines or data bytes with command fields.

A multi-master system may also be configured by the user. A system of this type is shown in Figure 8-1(b). An exchange of master control could be implemented using a handshake method

through the I/O ports or by an exchange of code messages through the serial peripheral interface system. There are two bits which are important to this configuration, the MSTR bit in the serial peripheral control register and the MODF bit in the serial peripheral status register.

9
ANALOG TO DIGITAL CONVERTER

9.1 Introduction

The A/D converter system consists of an 8-bit successive approximation converter and a 16-channel multiplexer. Four of the channels are available for output, and the other twelve channels are dedicated to internal test functions. There is one 8-bit result data register (address $28) and one 8-bit status/control register (address $29). The reference supply for the A/D converter uses dedicated input pins instead of the power supply lines, because drops caused by loading in the power supply lines would degrade the accuracy of the A/D conversion. An internal RC oscillator is available if the bus speed is low enough to degrade the A/D accuracy. An ADON bit allows the A/D to be switched off to reduce power consumption, which is particularly useful in the Wait mode.

9.2 Channel Assignment

Channel assignment is controlled by the A/D Status and Control register, ACR ($29). Refer to Table 9-1 for details. When the conversion is complete, the digital value is placed in the A/D Data register ($28); the conversion flag (COCO, bit 7 of ACR) is set; selected input is sampled again; and a new conversion begins.

The converter uses V_{RH} and V_{RL} as reference voltages. An input voltage equal to or greater than V_{RH} converts to $FF. An input voltage equal to or less than V_{RL}, but greater than V_{SS}, converts to $00. Maximum and minimum ratings must not be exceeded.To maintain full accuracy of the A/D, three requirements should be followed:

1) V_{RH} should be equal to or less than V_{CC},
2) V_{RL} should be equal to or greater than V_{SS} but less than maximum specifications, and
3) V_{RH}-V_{RL} should be equal to or greater than 4 Volts.

Table 9-1 A/D Channel Assignment

CH3	CH2	CH1	CH0	Channel Assignment
0	0	0	0	AN0 (port G bit 0)
0	0	0	1	AN1 (port G bit 1)
0	0	1	0	AN2 (port G bit 2)
0	0	1	1	AN3 (port G bit 3)
0	1	0	0	V_{RL} pin (low)
0	1	0	1	V_{RL} pin (low)
0	1	1	0	V_{RL} pin (low)
0	1	1	1	V_{RL} pin (low)
1	0	0	0	V_{RH} pin (high)
1	0	0	1	$(V_{RH}-V_{RL})\div 2$
1	0	1	0	V_{RL} pin (low)
1	0	1	1	V_{RL} pin (low)
1	1	0	0	V_{RL} pin (low)
1	1	0	1	V_{RL} pin (low)
1	1	1	0	V_{RL} pin (low)
1	1	1	1	V_{RL} pin (low)

9.3 A/D Status and Control Register

Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
$29	COCO	ADRC	ADON		CH3	CH2	CH1	CH0	000- 0000

A write to the A/D Status and Control register will cause the converter to abort the current conversion; reset the COCO flag and start a new conversion on the selected channel.

COCO - Conversion Complete

1 (set) – Conversion is complete; a new result can be read from the A/D Data register at address $28. The converter immediately starts a new conversion. This bit is cleared by reading the A/D Data register, writing to the A/D Status and Control register, power-on reset, or by an external reset.

0 (clear) – No conversion since last reset.

ADRC - A/D RC Oscillator Control

1 (set) – A/D uses RC clock.

0 (clear) – A/D uses CPU clock

When the oscillator is turned on, it requires a time t_{ADRC} to stabilize (see Table 13-5 and Table 13-6), and results can be inaccurate during this time. The ADRC bit allows the user to control the A/D RC oscillator, which is used to provide a sufficient high speed clock rate for the A/D when the MCU is running at low speeds. The A/D will take 32 bus cycles to complete a conversion. If internal RC clock is selected as the clock source, it will take 32µs to complete a conversion.

ADON - A/D On

1 (set) – A/D enabled

0 (clear) – A/D disabled

When the A/D is turned on, it requires a time t_{ADON} to stabilize (see Table 13-5 and Table 13-6) and results can be inaccurate during this time.

CH3-CH0 - A/D Channels 3 to Channel 0

These bits select the A/D channel assignment (see Table 9-1).

Note: Using one or more pins of PG0/AN0 - PG3/AN3 as analog inputs does not affect the ability to use port G inputs as digital inputs. However, using port G for digital inputs during analog conversion sequence may inject noise on the analog inputs and reduce the accuracy of the A/D conversion.
Performing a digital read of port G with levels other than V_{DD} or V_{SS} on the inputs causes greater than normal power dissipation during the read and may give erroneous results.

THIS PAGE LEFT BLANK INTENTIONALLY

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 MC68HC05G1.

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.

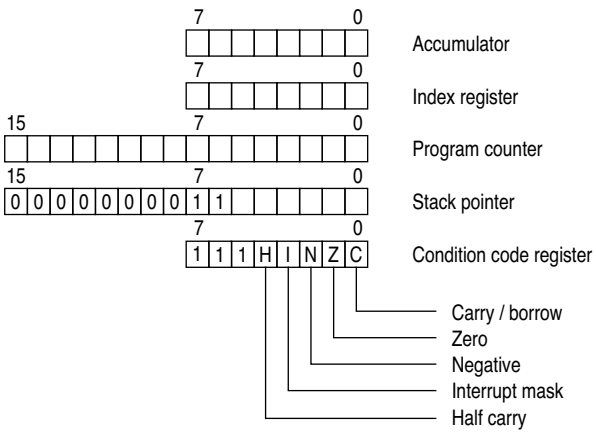


Figure 10-1 Programming model

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.

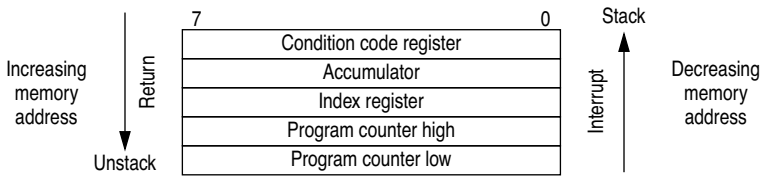


Figure 10-2 Stacking order

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

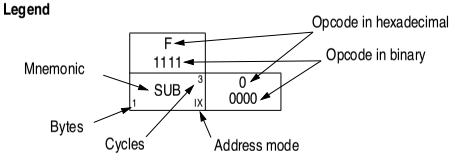
	Bit manipulation		Branch	Read/modify/write					Control		Register/memory						
	BTB	BSC	REL	DIR	INH	INH	IX1	IX	INH	INH	IMM	DIR	EXT	IX2	IX1	IX	
High / Low	0 0000	1 0001	2 0010	3 0011	4 0100	5 0101	6 0110	7 0111	8 1000	9 1001	A 1010	B 1011	C 1100	D 1101	E 1110	F 1111	High / Low
0 0000	BRSET0 5 3 BTB	BSET0 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 ***M6805 HMOS/M146805 CMOS Family Microcomputer/ Microprocessor User's Manual*** or 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 MC68HC05G1 has two low power modes, Stop and Wait. These two modes have different effects on the Programmable Timer, Real Time Clock, and Serial Peripheral Interface.

11.1 Stop Mode

The Stop mode is entered when the processor executes the STOP instruction. This places the MC68HC05G1 in its lowest power consumption mode. In Stop mode the internal processor clock is turned off, causing all internal processing to be halted (see Stop mode flowchart in Figure 11-1). Entering Stop mode, the I bit (interrupt mask) in the condition code register is cleared to enable external interrupts ($\overline{IRQ}$, $\overline{INT1}$ & $\overline{INT2}$) and RTC interrupt. All other registers and memory remain unaltered and all input/output lines remain unchanged. The RTC will keep running if the STPOSC=0 (32KHz oscillator kept running) in the PLL Status and Control register. Setting the STPOSC bit before entering Stop mode, the RTC will stop. Stop mode is exited by an external interrupt ($\overline{IRQ}$, $\overline{INT1}$ & $\overline{INT2}$), RTC interrupt, or external $\overline{RESET}$. On exit, the internal processor clock is then turned on and the MCU will wait for 16 cycles of 32KHz clock if the 32KHz oscillator has been kept running continuously. Otherwise the 32KHz oscillator should be turned on and, like Power-On-Reset, the MCU operation is halted for 4064 cycles of 32KHz clock to ensure a stable crystal oscillation. This latter case will happen if the interrupt is an external reset. The program counter is vectored to memory locations containing the start address of the interrupt or reset service routine.

The effects of Stop mode on the programmable Timer, RTC, and SPI are described below.

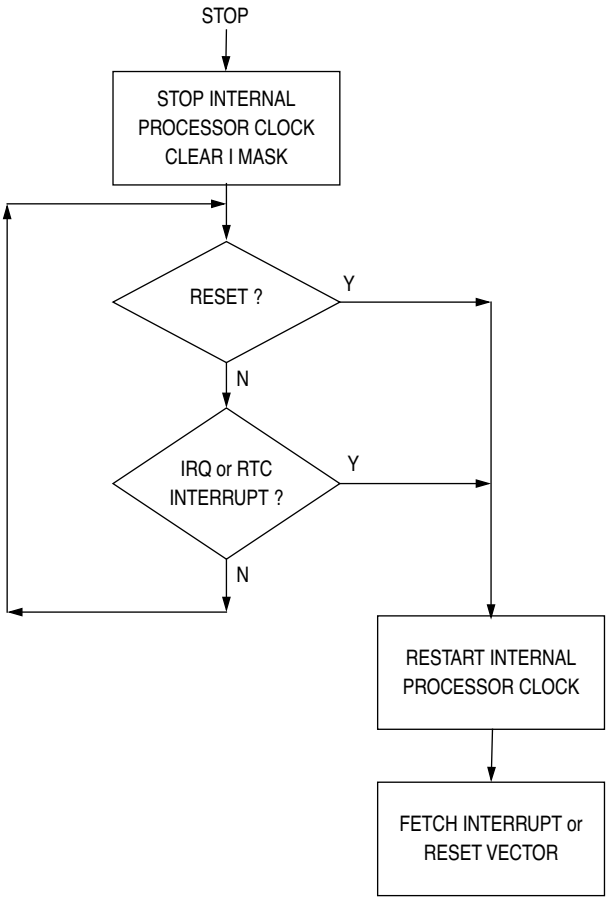


Figure 11-1 Stop Flowchart

11.1.1 Programmable Timer during Stop Mode

When the MCU enters Stop mode, the timer counter stops counting (the internal processor clock is stopped) and remains at that particular count value until the Stop mode is exited by an interrupt, the counter then resumes counting. If the Stop mode is exited by a reset the counter is forced to $FFFC. Another feature of the programmable timer in 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 Stop mode. If the Stop mode is exited by an external reset (logic low on $\overline{\text{RESET}}$ pin), then no such input capture flag or data action takes place even if there was a valid input capture edge (at the TCAP pin) during MCU Stop mode.

11.1.2 RTC during Stop Mode

The RTC runs at 32.768KHz. The Stop instruction has no effect on the RTC if the STPOSC bit in the PLL Status and Control register ($25) is cleared. If the STPOSC bit is set, the RTC stops operating in Stop mode because the 32.768KHz crystal oscillator is disabled.

11.1.3 SPI during Stop Mode

When the MCU enters Stop mode, the baud rate generator which drives the SPI shuts down. This essentially stops all master mode SPI operation, 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 Stop mode. If Stop mode is exited by a reset, then 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 and clock information in addition to transmitting its own data back to a master device.

At the end of a possible transmission with a slave SPI in Stop mode, no flags are set until a logic low on $\overline{\text{IRQ}}$, $\overline{\text{INT1}}$, or $\overline{\text{INT2}}$ input results in an MCU "wake-up". Caution should be observed when operating the SPI (as slave) during Stop mode because none of the protection circuitry (write collision, mode fault, etc.) is active.

It should also be noted that when the MCU enters the Stop mode all enabled output drivers (TDO, TCMP, MISO, MOSI, and SCK ports) remain active and any sourcing from these outputs will be part of the total supply current required by the device.

See also Section 8.6.

11.2 Wait Mode

The WAIT instruction places the MC68HC05G1 in a low power consumption mode. In Wait mode, all CPU activities are stopped, but the internal clock, programmable timer, serial peripheral interface and real time clock remain active. Refer to Figure 11-2 for Wait flowchart. On entering Wait mode, the I bit in the condition code register is cleared to enable all interrupts. In Wait mode all registers and memory remain unaltered and all parallel input/output lines remain unchanged. In additions to interrupts from $\overline{IRQ}$, $\overline{INT1}$, $\overline{INT2}$, RTC, interrupts from either programmable timer, or SPI will cause the processor to exit Wait mode. If a non-reset exit from Wait mode is performed (e.g. timer overflow interrupt exit), the system will continue from the state before it entered Wait mode. If a reset exit from Wait mode is performed all the systems revert to the disabled reset state.

In Wait mode, device power consumption depends on how many systems are active. The power consumption will be the lowest when the SPI and the Timer are disabled (the RTC cannot be disabled in Wait mode).

11.2.1 Programmable during Timer Wait Mode

The timer system is not affected by the Wait mode and continues regular operation. Any valid timer interrupt will wake the system up.

11.2.2 RTC during Wait Mode

The RTC system is not affected by the Wait mode and continues regular operation. Any valid RTC interrupt will wake the system up.

11.2.3 SPI during Wait Mode

The SPI system is not affected by the Wait mode and continues regular operation. Any valid SPI interrupt will wake the system up. See also Section 8.5.

11.3 Data Retention Modes

The contents of RAM and CPU registers are retained at supply voltages as low as 2.0 Vdc. This is referred to as the data retention mode, where the data is held, but the device is not guaranteed to operate.

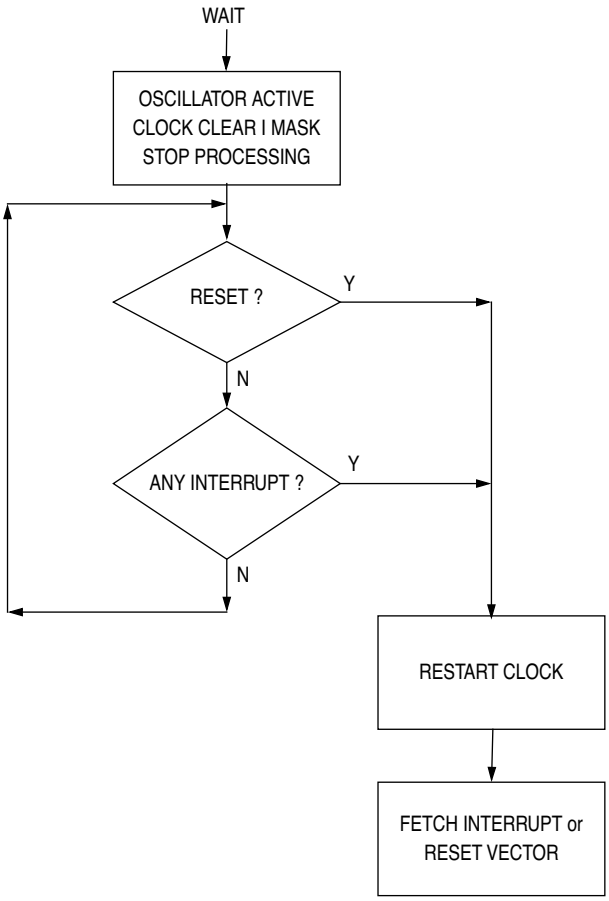


Figure 11-2 Wait Flowchart

THIS PAGE LEFT BLANK INTENTIONALLY

12
OPERATING MODES

The MC68HC05G1/*MC68HC705G1* 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.

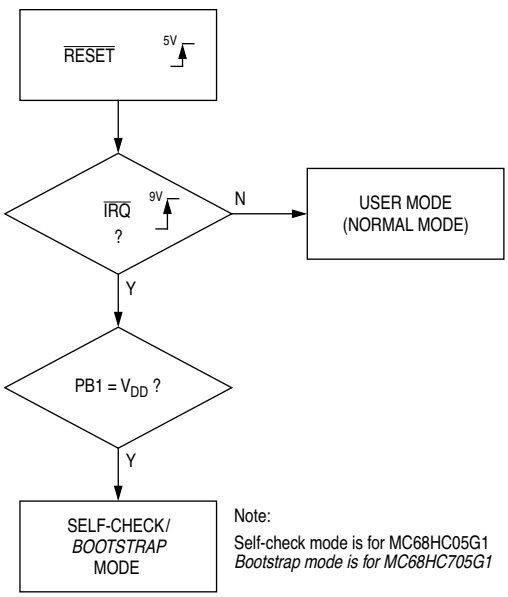


Figure 12-1 Flowchart of Mode Entering

Table 12-1 Mode Selection

$\overline{\text{RESET}}$	$\overline{\text{IRQ}}$	PB1	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 $\overline{\text{IRQ}}$ function pin.			

12.1 User Mode (Normal Operation)

The normal operational mode of the MC68HC05G1/*MC68HC705G1* is the user mode. The user mode will be entered if the $\overline{\text{RESET}}$ line is brought low, and the $\overline{\text{IRQ}}$ pin is within its normal operational 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 MC68HC05G1 self-check mode is for the user to check device functions with an on-chip self-check program masked at location $3E00 to $3FDF under minimum hardware support. The hardware is shown in Figure 12-3. Figure 12-2 is the criteria to enter self-check mode, where PB1's condition is latched within first two clock cycles after the rising edge of the reset. PB1 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 can show what part of the device is suspected to be bad. Table 12-2 lists the LEDs' on-off patterns and their corresponding indications.

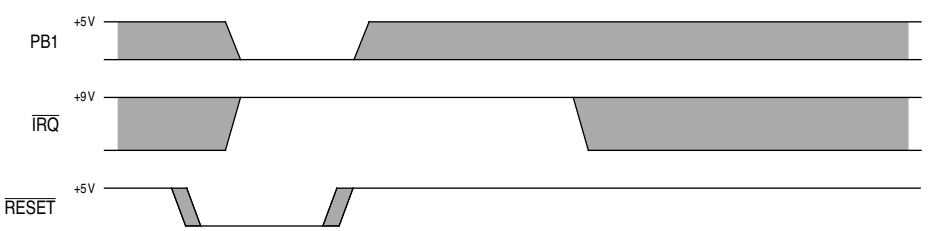


Figure 12-2 Self-Check Mode Timing

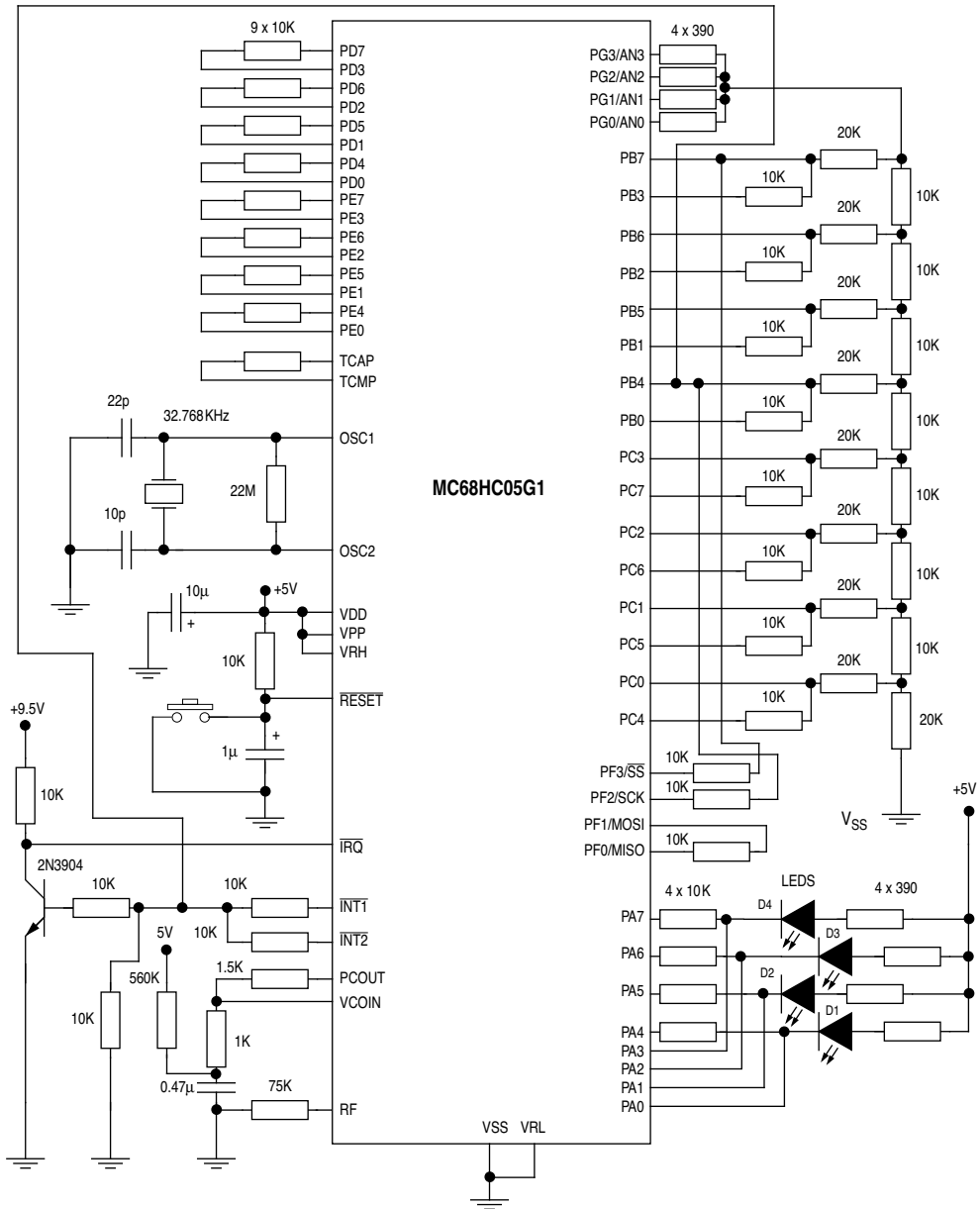


Figure 12-3 MC68HC05G1 Self-Test Circuit

12

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
1	0	1	0	Bad A/D
1	0	0	1	Bad SPI
1	0	0	0	Bad RTC
0	1	1	1	Bad Interrupts
Flashing				Good Device
All Others				Bad Device, Bad Port A, etc.

1=LED off; 0=LED on.

12.3 Bootstrap Mode

The bootstrap mode is provided in the EPROM part (MC68HC705G1) as a mean of self-programming its EPROM with minimal circuitry. It is entered on the rising edge of $\overline{\text{RESET}}$ if $\overline{\text{IRQ}}$ pin is at $1.8V_{DD}$ and PB1 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 LED1 and LED2. The EPROM programming circuit for bootstrap mode is shown in Figure 12-4.

Table 12-3 Bootstrap Mode Options

PD3	PD2	PD1	PD0	LED1	LED2	REMARKS
0	0	0	1	Off	Verify	Verify
0	0	1	1	Program	Verify	Program & Verify
0	1	0	1	Off	Verify	Blank EPROM check

12.3.1 Programming Control Register (PCR)

Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
$3D							ELAT	PGM	---- --00

ELAT - EPROM Latch Control

1 (set) – EPROM address and data bus configured for programming (writes to EPROM cause address data to be latched). EPROM is in programming mode and cannot be read if ELAT is 1. This bit is not able be set when no V_{PP} voltage is applied to the V_{PP} pin.

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

PGM - EPROM Program Command

1 (set) – Programming power switched on to EPROM array. If ELAT not = 1 then PGM = 0.

0 (clear) – Programming power switched off to EPROM array.

12.3.2 EPROM Programming Sequence

Programming the EPROM of the MC68HC705G1 is as follows:

1) Set the ELAT bit.
2) Write the data to be programmed to the address to be programmed.
3) Set the PGM bit.
4) Delay for the appropriate amount of time.
5) Clear the PGM and the ELAT bits.

The last action may be carried out in a single CPU write operation. It is important to remember that an external programming voltage must be applied to the V_{PP} pin while programming, but should be equal to V_{DD} during normal operation.

Example shows address $2000 is programmed with $00.

```
CLR    PCR          ;reset PCR
LDX    #$00         ;load index register with 00
BSET   1,PCR        ;set ELAT bit
LDA    #$00         ;load data=00 in to A
STA    $2000,X      ;latch data and address
BSET   0,PCR        ;program
JSR    DELAY        ;call delay subroutine
CLR    PCR          ;reset PCR
```

Please refer to *MC68HC705G1PGMR PROGRAMMING BOARD, User's Guide rev.10* for more details.

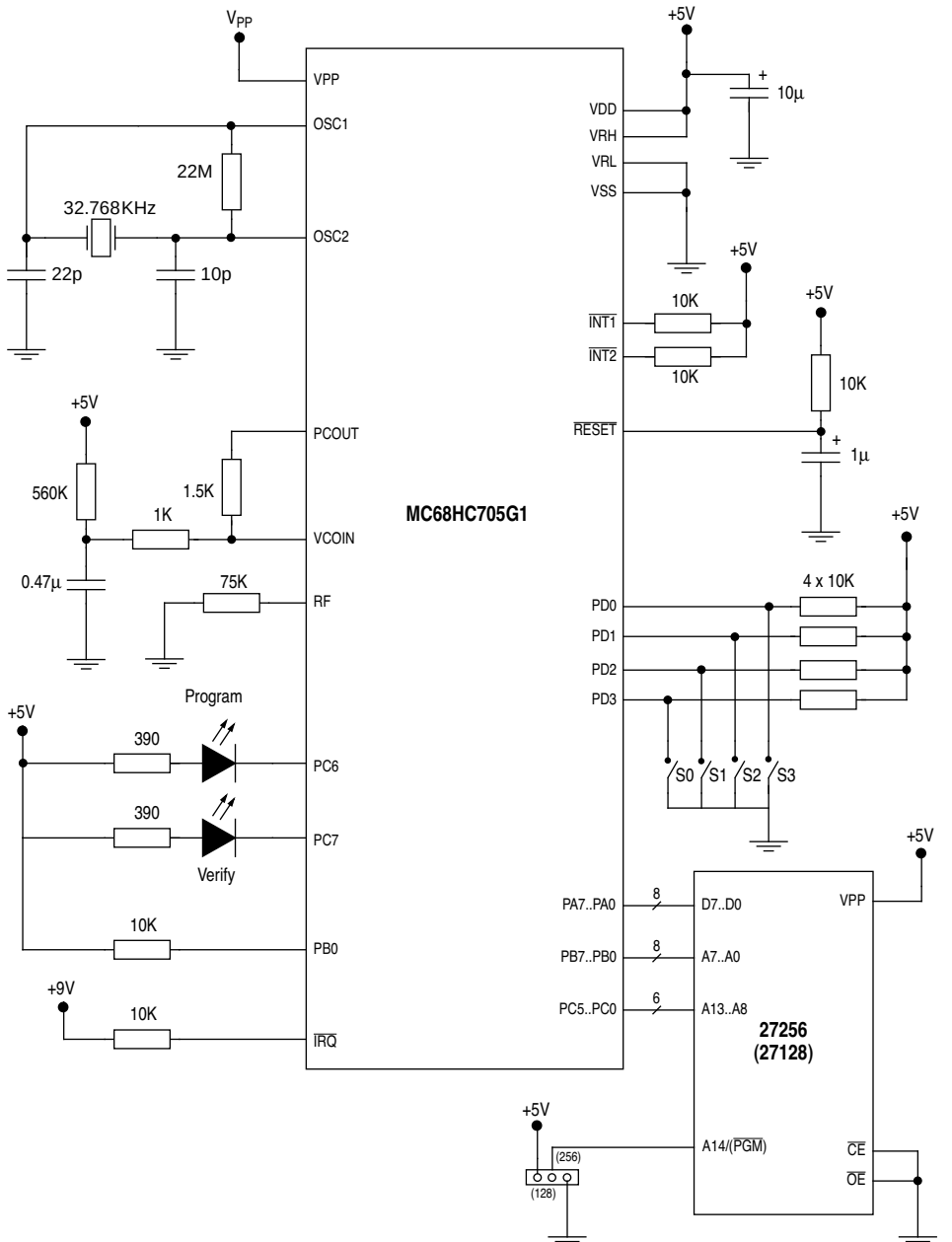


Figure 12-4 EPROM Programming Circuit for Bootstarp mode

13
ELECTRICAL SPECIFICATIONS

This section contains the electrical specifications for MC68HC05G1/*MC68HC705G1*.

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 $V_{SS} \le (V_{in} \text{ or } V_{out}) \le V_{DD}$. 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 - Plastic 64-pin QFP package	 θ_{JA} θ_{JA}	 50 50	 °C/W °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} = –10µA	V_{OH}	V_{DD}-0.1	–	–	V
I_{LOAD} = +10µA	V_{OL}	–	–	0.1	V
Output high voltage (I_{LOAD}=–0.8mA) PA0-PA7, PB0-PB7, PC0-PC7, PD1-PD7, PE0-PE7, PF0-PF2, TCMP	V_{OH}	V_{DD}-0.8	–	–	V
Output low voltage (I_{LOAD}=+1.6mA) PA0-PA7, PB0-PB7, PC0-PC7, PD1-PD7, PE0-PE7, PF0-PF2, TCMP	V_{OL}	–	–	0.4	V
Input high voltage PA0-PA7, PB0-PB7, PC0-PC7, PD0-PD7, PE0-PE7, PF0-PF3, PG0-PG3, TCAP, $\overline{IRQ}$, $\overline{INT1}$, $\overline{INT2}$, $\overline{RESET}$, OSC1	V_{IH}	0.7xV_{DD}	–	V_{DD}	V
Input low voltage PA0-PA7, PB0-PB7, PC0-PC7, PD0-PD7, PE0-PE7, PF0-PF3, PG0-PG3, TCAP, $\overline{IRQ}$, $\overline{INT1}$, $\overline{INT2}$, $\overline{RESET}$, OSC1	V_{IL}	V_{SS}	–	0.2xV_{DD}	V
Data Retention Mode	V_{RM}	2.0	–	–	V
Supply current:					
Run (32KHz)		–	1.0	–	mA
(0.164MHz)		–	1.1	–	mA
(0.328MHz)		–	1.3	–	mA
(1.311MHz)		–	2.8	–	mA
(2.622MHz)		–	5.0	8.0	mA
Wait (32KHz)	I_{DD}	–	0.4	–	mA
(0.164MHz)		–	0.5	–	mA
(0.328MHz)		–	0.6	–	mA
(1.311MHz)		–	0.9	–	mA
(2.622MHz)		–	1.3	2.5	mA
Stop (32KHz oscillator on)		–	4	10	µA
(32KHz oscillator off)		–	4	10	µA
I/O ports high-Z leakage current PA0-PA7, PB0-PB7, PC0-PC7, PD0-PD7, PE0-PE7, PF0-PF3	I_{IL}	–	–	±10	µA
Input current					
TCAP, $\overline{IRQ}$, $\overline{INT1}$, $\overline{INT2}$, $\overline{RESET}$, PF0-PF3, OSC1	I_{IN}	–	–	±1	µA
PG0-PG3 (A/D off)		–	±0.2	±1	µA
PG0-PG3 (A/D on)		–	±10	–	µA
Capacitance					
ports (as input or output), $\overline{RESET}$, $\overline{IRQ}$,	C_{OUT}	–	–	12	pF
$\overline{INT1}$, $\overline{INT2}$, TCAP, OSC1, PF0-PF3	C_{IN}	–	–	8	pF
PG0-PG3 (A/D off)	C_{IN}	–	12	–	pF
PG0-PG3 (A/D on)	C_{IN}	–	22	–	pF

Table 13-2 DC Electrical Characteristics for 3.3V operation

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

CHARACTERISTICS	SYMBOL	MINIMUM	TYPICAL	MAXIMUM	UNIT
Output voltage					
I_{LOAD} = –10µA	V_{OH}	V_{DD}-0.1	–	–	V
I_{LOAD} = +10µA	V_{OL}	–	–	0.1	V
Output high voltage (I_{LOAD}=–0.3mA) PA0-PA7, PB0-PB7, PC0-PC7, PD1-PD7, PE0-PE7, PF0-PF2, TCMP	V_{OH}	V_{DD}-0.3	–	–	V
Output low voltage (I_{LOAD}=+0.3mA) PA0-PA7, PB0-PB7, PC0-PC7, PD1-PD7, PE0-PE7, PF0-PF2, TCMP	V_{OL}	–	–	0.3	V
Input high voltage PA0-PA7, PB0-PB7, PC0-PC7, PD0-PD7, PE0-PE7, PF0-PF3, PG0-PG3, TCAP, $\overline{IRQ}$, $\overline{INT1}$, $\overline{INT2}$, $\overline{RESET}$, OSC1	V_{IH}	0.7xV_{DD}	–	V_{DD}	V
Input low voltage PA0-PA7, PB0-PB7, PC0-PC7, PD0-PD7, PE0-PE7, PF0-PF3, PG0-PG3, TCAP, $\overline{IRQ}$, $\overline{INT1}$, $\overline{INT2}$, $\overline{RESET}$, OSC1	V_{IL}	V_{SS}	–	0.2xV_{DD}	V
Data Retention Mode	V_{RM}	2.0	–	–	V
Supply current:					
Run (32KHz)		–	0.8	–	mA
(0.164MHz)		–	0.9	–	mA
(0.328MHz)		–	1.0	–	mA
(1.311MHz)		–	1.7	4	mA
Wait (32KHz)	I_{DD}	–	0.35	–	mA
(0.164MHz)		–	0.4	–	mA
(0.328MHz)		–	0.45	–	mA
(1.311MHz)		–	0.6	1.5	mA
Stop (32KHz oscillator on)		–	2.5	6	µA
(32KHz oscillator off)		–	2.5	6	µA
I/O ports high-Z leakage current PA0-PA7, PB0-PB7, PC0-PC7, PD0-PD7, PE0-PE7, PF0-PF3	I_{IL}	–	–	±10	µA
Input current					
TCAP, $\overline{IRQ}$, $\overline{INT1}$, $\overline{INT2}$, $\overline{RESET}$, PF0-PF3, OSC1	I_{IN}	–	–	±1	µA
PG0-PG3 (A/D off)	I_{IN}	–	±0.2	±1	µA
PG0-PG3 (A/D on)	I_{IN}	–	±10	–	µA
Capacitance					
ports (as input or output), $\overline{RESET}$, $\overline{IRQ}$,	C_{OUT}	–	–	12	pF
$\overline{INT1}$, $\overline{INT2}$, TCAP, OSC1, PF0-PF3	C_{IN}	–	–	8	pF
PG0-PG3 (A/D off)	C_{IN}	–	12	–	pF
PG0-PG3 (A/D on)	C_{IN}	–	22	–	pF

13.4 A/D Converter Electrical Characteristics

Table 13-3 A/D Characteristics for 5V Operation

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

CHARACTERISTICS	PARAMETER	MINIMUM	MAXIMUM	UNIT
Resolution	Number of bits resolved by the ADC	8	–	bits
Non-linearity	Maximum deviation from the best straight line through the ADC transfer characteristics (V_{RH}=V_{DD} and V_{RL}=0V)	–	±1/2	LSB
Quantization error	Uncertainty due to converter resolution	–	±1/2	LSB
Absolute accuracy	Difference between the actual input voltage and the full-scale equivalent of the binary code output code for all errors	–	±1	LSB
Conversion range	Analog input voltage range	V_{RL}	V_{RH}	V
V_{RH}	Maximum analog reference voltage	V_{RL}	V_{DD}+0.1	V
V_{RL}	Minimum analog reference voltage	V_{SS}-0.1	V_{RH}	V
Conversion time	Total time to perform a single analog to digital conversion (a) External clock (OSC1, OSC2) (b) Internal RC oscillator	 – –	 32 32	 t_{CYC} µs
Monotonicity	Conversion result never decreases with an increase in input voltage and has no missing codes	Guaranteed		
Zero-input reading	Conversion result when V_{IN}=V_{RL}	00	–	hex
Full-scale reading	Conversion result when V_{IN}=V_{RH}	–	FF	hex
Sample acquisition time	Analog input acquisition sampling (a) External clock (OSC1, OSC2) (b) Internal RC oscillator(1)	 – –	 12 12	 t_{CYC} µs
Sample/hold capacitance	Input capacitance on PG0/AN0-PG3/AN3	–	12	pF
Input leakage(2)	Input leakage on A/D pins PG0/AN0-PG3/AN3, V_{RL}, V_{RH}	1	10	µA

Notes:

(1) Source impedances greater than 10KΩ will adversely affect internal charging time during input sampling.

(2) The external system error caused by input leakage current is approximately equal to the product of R source and input current.

Table 13-4 A/D Characteristics for 3.3V Operation

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

CHARACTERISTICS	PARAMETER	MINIMUM	MAXIMUM	UNIT
Resolution	Number of bits resolved by the ADC	8	–	Bit
Non-linearity	Maximum deviation from the best straight line through the ADC transfer characteristics (V_{RH}=V_{DD} and V_{RL}=0V)	–	±1	LSB
Quantization error	Uncertainty due to converter resolution	–	±1	LSB
Absolute accuracy	Difference between the actual input voltage and the full-scale equivalent of the binary code output code for all errors	–	±2	LSB
Conversion range	Analog input voltage range	V_{RL}	V_{RH}	V
V_{RH}	Maximum analog reference voltage	V_{RL}	V_{DD}+0.1	V
V_{RL}	Minimum analog reference voltage	V_{SS}-0.1	V_{RH}	V
Conversion time	Total time to perform a single analog to digital conversion Internal RC oscillator	–	32	µs
Monotonicity	Conversion result never decreases with an increase in input voltage and has no missing codes	Guaranteed		
Zero-input reading	Conversion result when V_{IN}=V_{RL}	00	–	Hex
Full-scale reading	Conversion result when V_{IN}=V_{RH}	–	FF	Hex
Sample acquisition time	Analog input acquisition sampling Internal RC oscillator(1)	–	12	µs
Sample/hold capacitance	Input capacitance on PG0/AN0-PG3/AN3	–	12	pF
Input leakage(2)	Input leakage on A/D pins PG0/AN0-PG3/AN3, V_{RL}, V_{RH}	1	10	µA

Notes:

(1) Source impedances greater than 10KΩ will adversely affect internal charging time during input sampling.

(2) The external system error caused by input leakage current is approximately equal to the product of R source and input current.

13.5 Control Timing

Table 13-5 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	4.2 4.2	MHz MHz
Internal operating frequency (f_{OSC}/2) Crystal External clock	 f_{OP} f_{OP}	 – dc	 2.1 2.1	 MHz MHz
Processor cycle time	t_{CYC}	480	–	ns
Crystal oscillator start-up time	t_{OXOV}	–	100	ms
Stop recovery start-up time (crystal oscillator)	t_{ILCH}		100	ms
RC oscillator stabilization time	t_{ADRC}		100	ms
A/D converter stabilization time	t_{ADON}		500	µs
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}	2^{20}	–	t_{CYC}
Timer Resolution(1) Input capture pulse width Input capture pulse period	 t_{RESL} t_{TH}, t_{TL} t_{TLTL}	 4 125 –(2)	 – – –	 t_{CYC} ns t_{CYC}
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 13-6 Control Timing for 3.3V 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	2.0 2.0	MHz MHz
Internal operating frequency (f_{OSC}/2) Crystal External clock	 f_{OP} f_{OP}	 – dc	 1.0 1.0	 MHz MHz
Processor cycle time	t_{CYC}	1000	–	ns
Crystal oscillator start-up time	t_{OXOV}	–	100	ms
Stop recovery start-up time (crystal oscillator)	t_{ILCH}		100	ms
RC oscillator stabilization time	t_{ADRC}		100	ms
A/D converter stabilization time	t_{ADON}		500	µs
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}	2^{20}	–	t_{CYC}
Timer Resolution(1) Input capture pulse width Input capture pulse period	 t_{RESL} t_{TH}, t_{TL} t_{TLTL}	 4 250 –(2)	 – – –	 t_{CYC} ns t_{CYC}
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}.

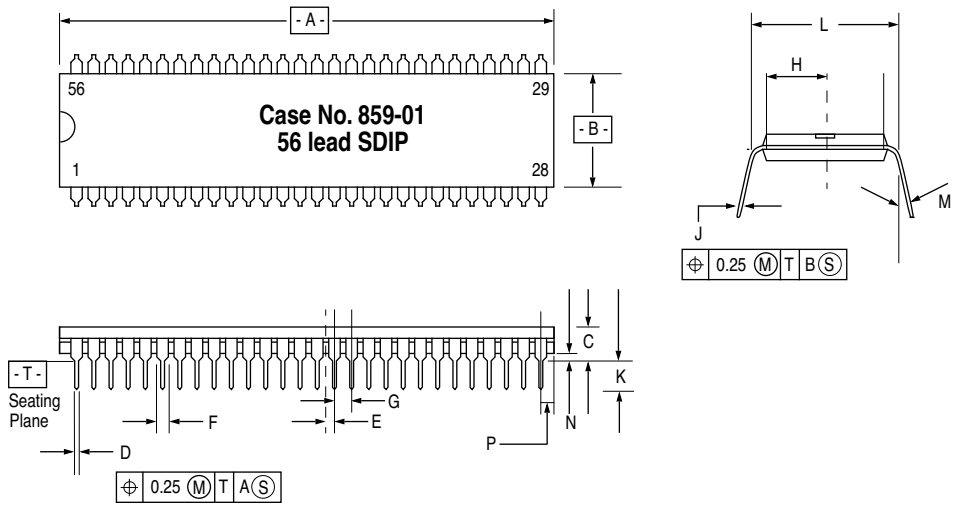
(3) The minimum period t_{ILIL} should not be less than the number of cycle times it takes to execute the interrupt service routine plus 21 t_{CYC}.

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 MC68HC05G1/*MC68HC705G1*.

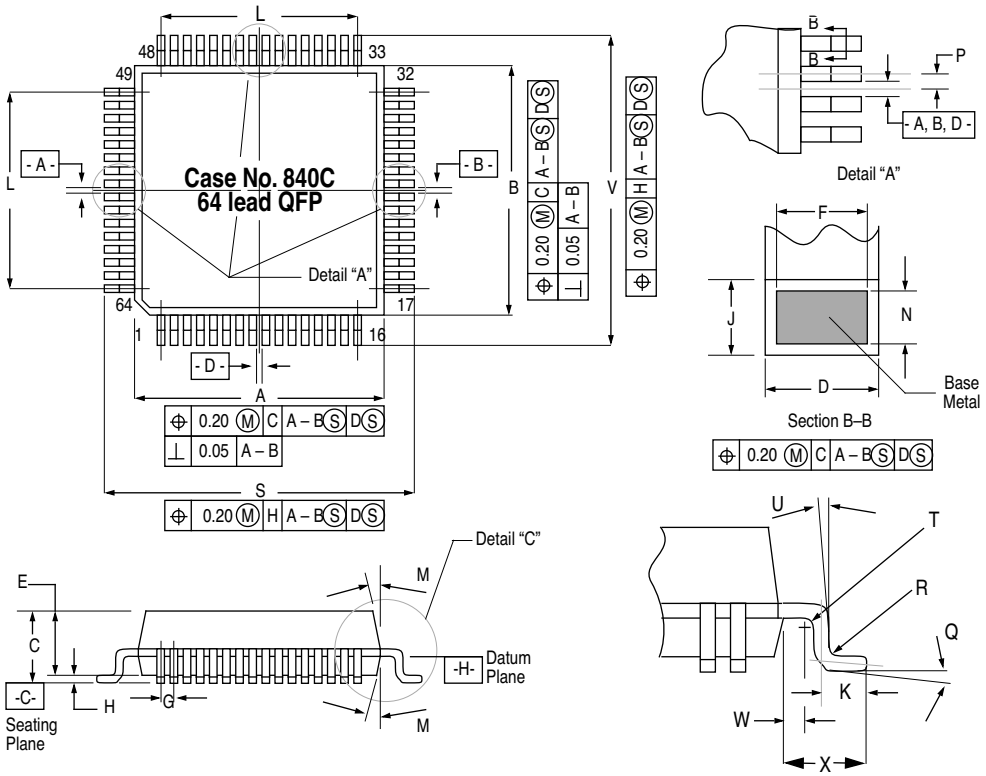
14.1 56-pin SDIP Package



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 Dimension

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 Dimemsions

Literature Distribution Centres:

EUROPE: Motorola Ltd., European Literature Centre, 88 Tanners Drive, Blakelands, Milton Keynes, MK14 5BP, England.

ASIA PACIFIC: Motorola Semiconductors (H.K.) Ltd., Silicon Harbour Center, No. 2, Dai King Street, Tai Po Industrial Estate, Tai Po, N.T., Hong Kong.

JAPAN: Nippon Motorola Ltd., 4-32-1, Nishi-Gotanda, Shinagawa-ku, Tokyo 141, Japan.

USA: Motorola Literature Distribution, P.O. Box 20912, Phoenix, Arizona 85036.