MC68HC05BS8D/H

C68HC05BS8

HC05

TECHNICAL DATA

MC68HC05BS8
MC68HC705BS8

TECHNICAL
DATA

freescale semiconductor

MC68HC05BS8
MC68HC705BS8

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

Conventions

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

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

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

CUSTOMER FEEDBACK QUESTIONNAIRE (MC68HC05BS8D/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	□	Other □	Please specify: ____________________
System design	□		____________________
Training purposes	□		____________________

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 DESCRIPTION	□	□	□	□	□
SECTION 3	INPUT/OUTPUT PORTS	□	□	□	□	□
SECTION 4	MEMORY AND REGISTERS	□	□	□	□	□
SECTION 5	RESETS AND INTERRUPTS	□	□	□	□	□
SECTION 6	TIMERS	□	□	□	□	□
SECTION 7	PULSE WIDTH MODULATION	□	□	□	□	□
SECTION 8	M-BUS SERIAL INTERFACE	□	□	□	□	□
SECTION 9	SYNC SIGNAL PROCESSOR	□	□	□	□	□
SECTION 10	CPU CORE AND INSTRUCTION SET	□	□	□	□	□
SECTION 11	LOW POWER MODES	□	□	□	□	□
SECTION 12	OPERATING MODES	□	□	□	□	□
SECTION 13	ELECTRICAL SPECIFICATIONS	□	□	□	□	□
SECTION 14	MECHANICAL SPECIFICATIONS	□	□	□	□	□
APPENDIX A	MC68HC705BS8	□	□	□	□	□

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

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: MC68HC05BS8D/H)

FAX: (852) 2485-0548

MOTOROLA
Semiconductor Products Sector
Asia Pacific Group

– First fold back along this line –

– Cut along this line to remove –

– 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: ______________

____

or helping us improve our documentation,
echnical Publications , Motorola Semiconductors H.K. Ltd., Hong Kong.

– Finally, tuck this edge into opposite flap –

TABLE OF CONTENTS

Paragraph Number — TITLE — Page Number

1
GENERAL DESCRIPTION

2
PIN DESCRIPTION

3
INPUT/OUTPUT PORTS

4
MEMORY AND REGISTERS

Paragraph Number TITLE Page Number

5
RESETS AND INTERRUPTS

6
TIMERS

7
PULSE WIDTH MODULATION

8
M-BUS SERIAL INTERFACE

9
SYNC SIGNAL PROCESSOR

Paragraph Number | TITLE | Page Number

10
CPU CORE AND INSTRUCTION SET

11
LOW POWER MODES

Paragraph Number | TITLE | Page Number

12
OPERATING MODES

13
ELECTRICAL SPECIFICATIONS

14
MECHANICAL SPECIFICATIONS

A
MC68HC705BS8

THIS PAGE LEFT BLANK INTENTIONALLY

LIST OF FIGURES

Figure Number **TITLE** Page Number

LIST OF TABLES

THIS PAGE LEFT BLANK INTENTIONALLY

1
GENERAL DESCRIPTION

The MC68HC05BS8 HCMOS microcontroller is a member of the MC68HC05 Family of low-cost single-chip microcontrollers. It is particularly suitable as a multi-sync computer monitor controller. This 8-bit microcontroller unit (MCU) contains on-chip oscillator, CPU, RAM, ROM, EEPROM, I/O, Timers, COP Watchdog, M-Bus Serial Interface System, PWM, and Sync Signal Processor.

The MC68HC705BS8 is an EPROM version of the MC68HC05BS8. All references to the MC68HC05BS8 apply equally to the MC68HC705BS8, unless otherwise stated. *References specific to the MC68HC705BS8 are italicized in the text, and also, for quick reference, they are summarized in Appendix A.*

1.1 Features

- Fully static chip design featuring the industry standard 8-bit M68HC05 core
- Power saving Stop and Wait modes
- 256 bytes of RAM (64 bytes for stack)
- 10K-bytes of ROM for MC68HC05BS8
  10K-bytes of EPROM for MC68HC705BS8
- 512 bytes of EEPROM
- 24 bidirectional I/O lines
- 6 keyboard interrupts
- Core timer with RTI and COP watchdog reset
- 2 M-Bus ($I^2C^\dagger$) Serial Interfaces (one full H/W, one S/W with hardware support)
- Single channel 6-bit general purpose PWM
  Single channel 7-bit raster positioning PWM
- 16-bit programmable timer with one TCAP and one TCMP
- Sync signal processor

† I^2C-bus is a proprietary Philips interface bus

- Low voltage reset (LVR)
- Available in 44-pin QFP package

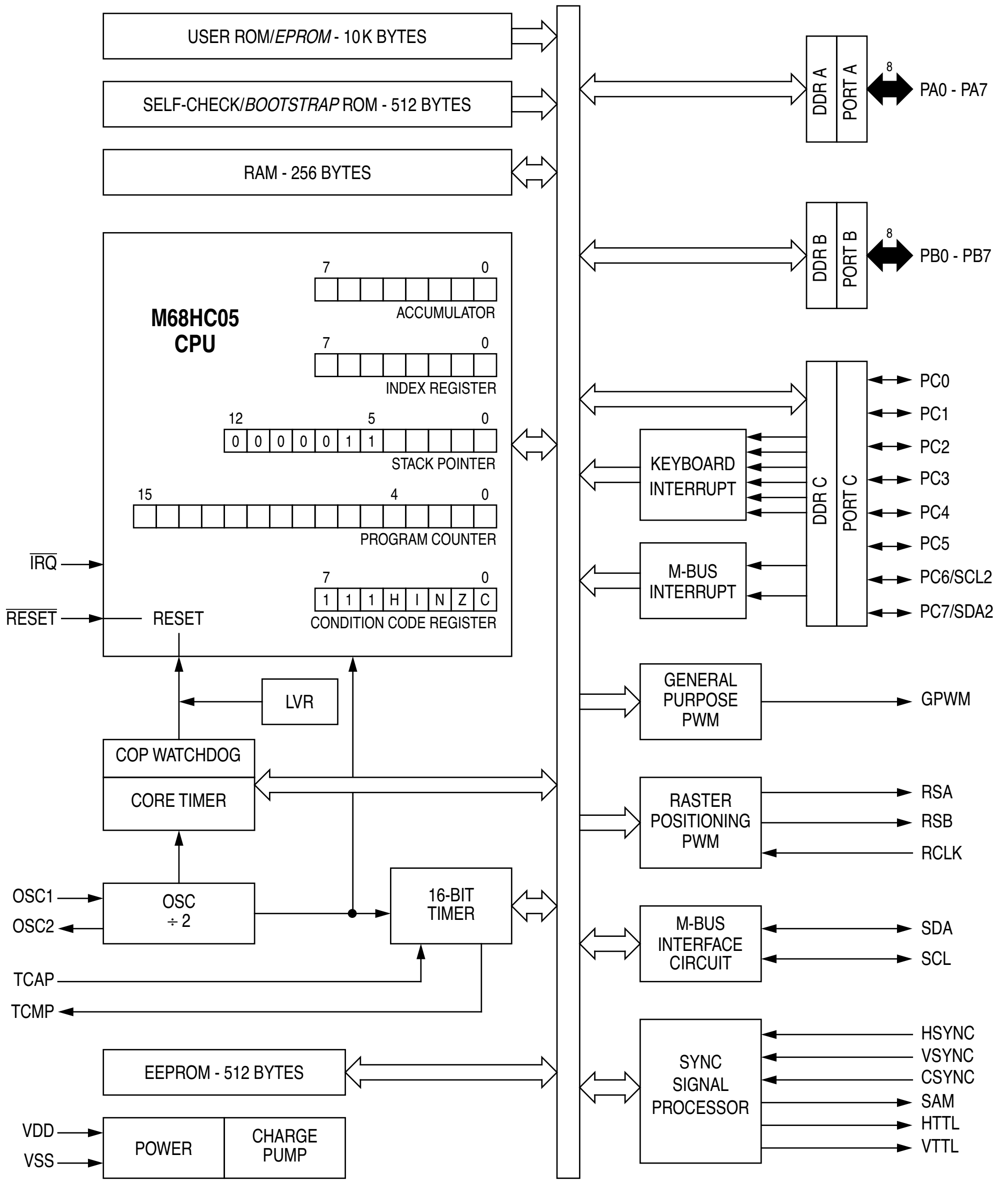
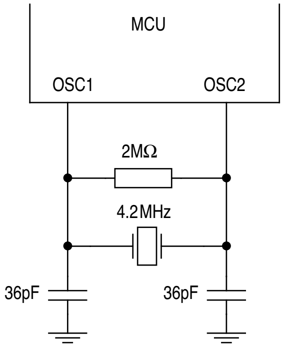


Figure 1-1 MC68HC05BS8/*MC68HC705BS8* Block Diagram

2
PIN DESCRIPTION

This section provides a description of the functional pins of the MC68HC05BS8 microcontroller.

2.1 Pin Descriptions

PIN NAME	44-pin QFP PIN No.	DESCRIPTION
VDD, VSS	38, 39	Power is supplied to the MCU using these pins. VDD is the positive power supply; VSS is ground.
$\overline{\text{IRQ}}$/ *VPP*	34	In the user mode this pin is the external hardware interrupt $\overline{\text{IRQ}}$. Two choices of interrupt triggering sensitivity are available through the Option register: 1) negative-edge sensitive triggering, or 2) negative-level sensitive triggering. *In bootstrap mode on the MC68HC705BS8, this is the EPROM programming voltage input pin.*
$\overline{\text{RESET}}$	35	The active low $\overline{\text{RESET}}$ 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.
OSC1, OSC2	36, 37	These pins provide connections to the on-chip oscillator. The oscillator can be driven by an AT-crystal circuit or a ceramic resonator with a maximum frequency of 4.4MHz. OSC1 may also be driven by an external oscillator if an external crystal/resonator circuit is not used. Example showing crystal connections. 
PA0-PA7	33-26	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	44-pin QFP PIN No.	DESCRIPTION
PB0-PB7	25-18	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. PB7 is also used as the RSPWM counter reset input when PB7 is set as an input pin and the counter reset enable bit is set in the GPWM register (bit 6 of $0010).
PC0-PC7	17-10	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. PC0-PC5 become keyboard interrupt input pins when the corresponding bits are set in the Keyboard Interrupt register ($001E). PC6 and PC7 are SDA and SCL respectively, when used for the software supported M-Bus Interface.
SDA, SCL	2, 3	These are the hardware M-Bus interface data and clock lines.
TCAP	9	This input pin controls the input capture function of the 16-bit free-running timer.
TCMP	8	This output pin indicates when a timer compare is successful.
GPWM	7	This is the output pin of the General purpose PWM.
RSA, RSB	5, 6	These are the two excursive outputs of the Raster Positioning PWM
RCLK	4	This is the input clock to drive the RSPWM counter.
HSYNC, VSYNC	40, 41	These two input pins are for the video sync signals from the host computer.
CSYNC	42	This is the Composite sync signal input from the host computer.
SAM	43	This is the output of an sample signal from the Sync Signal Processor.
HTTL, VTTL	44, 1	These are the output from the HSYNC and VSYNC inputs or the signals separated from CSYNC input.

2.2 Pin Assignment

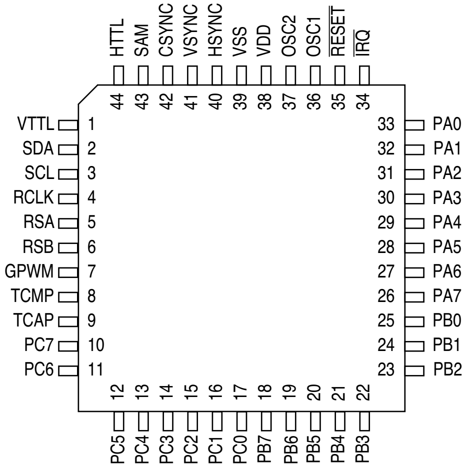


Figure 2-1 Pin Assignment for 44-pin QFP Package

3
INPUT/OUTPUT PORTS

The MC68HC05BS8 has 24 I/O lines, arranged as three 8-bit ports (Port A, B, and C). Each I/O line is individually programmable as either input or output, under the software control of the Data Direction registers. Port C also shares with keyboard interrupt and the software supported M-Bus functions.

To avoid glitches on the output pins, data should be written to the I/O Port Data register before writing “1”s to the corresponding Data Direction register bits to set the pins to output mode.

3.1 Input/Output Programming

Bidirectional port lines may be programmed as an input or an output under software control. The direction of the pins is determined by the state of the corresponding bit in the port data direction register (DDR). Each port has an associated DDR. Any I/O port 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, configuring all port pins as inputs. The data direction registers are capable of being written to or read by the MCU. 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. The operation of the standard port hardware is shown schematically in Figure 3-1.

This is summarized in Table 3-1 which shows the effect of reading from or writing to an I/O pin in various circumstances.

Table 3-1 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.

Note: $R/\overline{W}$ is an internal signal.

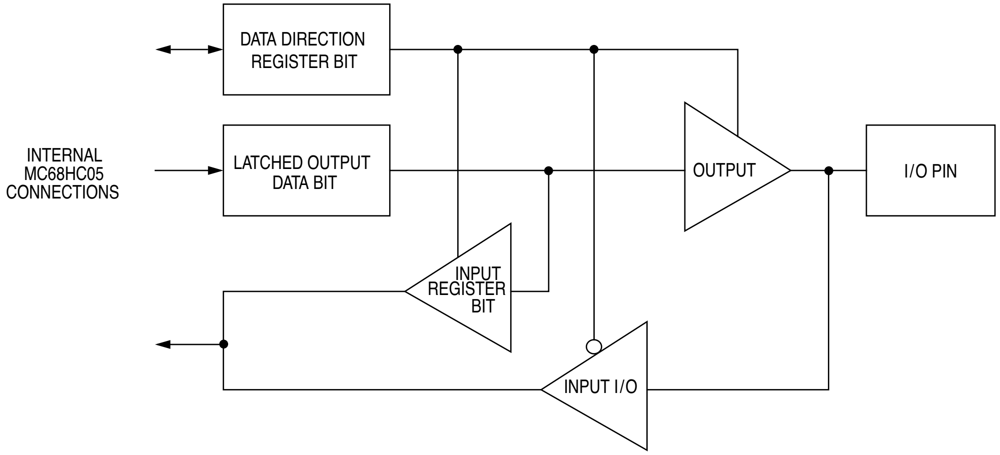


Figure 3-1 Port I/O Circuitry

3.2 Port A, B, and C

These are standard M68HC05 bidirectional I/O ports, each comprising a data register and a data direction register. All three are 8-bit ports.

Reset does not affect the state of the data registers, but clears the data direction registers, thereby returning all port pins to input mode. Writing a “1” to any DDR sets the corresponding port pin to output mode.

3.3 PB7 - RSPWM Counter Reset

In addition to normal I/O function, PB7 can be software selectable to input a reset signal to the Raster Positioning Pulse Width Modulator counter. The reset pulse requires an active high signal.

3.4 PC0:5 - Keyboard Interrupts

Six keyboard interrupt inputs are available on port pins PC0 to PC5. Each pin is enabled for keyboard interrupt by setting the corresponding keyboard interrupt enable bit in the KBI register (bits 0 to 5 of $001E). When the KBI bit is set, the corresponding port C pin will be configured as an input pin, regardless of the DDRC setting, an internal pull-up resistor is connected to the pin.

The interrupt signal is latched, and it should be cleared by writing a “1” to the KBIC bit in the KBI register (bit 6 of $001E) in the interrupt service routine. This should be cleared after the key is debounced, otherwise unwanted keyboard interrupt signals may be generated.

The keyboard interrupt is negative-edge sensitive only, and the interrupt service routine is specified by the contents of the memory locations $3FF0 and $3FF1.

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
KBI Register	$001E		KBIC	KBE5	KBE4	KBE3	KBE2	KBE1	KBE0	0000 0000

KBIC - Keyboard Interrupt Clear

1 (set) – Clear keyboard interrupt latch.

KBE5:0 - Keyboard Interrupt Enable 5:0

1 (set) – Enable keyboard interrupt for the corresponding bit.

0 (clear) – Disable keyboard interrupt for the corresponding bit.

3.5 PC6 and PC7 - Software Supported M-Bus SDA and SCL

PC6 and PC7 are used for SDA and SDL respectively when configured for the software supported M-Bus interface. This M-Bus interface is operated by software emulation, with hardware interrupt circuit connected to PC6 and PC7.

THIS PAGE LEFT BLANK INTENTIONALLY

4
MEMORY AND REGISTERS

The MC68HC05BS8/*MC68HC705BS8* has a 16K-byte memory map consisting of I/O registers, user ROM/*EPROM*, user RAM, EEPROM, self-check/*bootstrap* ROM as shown in Figure 4-1.

4.1 Registers

All the I/O, control and status registers of the MC68HC05BS8 are located within the first 64-byte block of the memory map (address $0000 to $003F).

4.2 RAM

The user RAM consists of 256 bytes of memory, from $0040 to $013F. This is shared with a 64 byte stack area. The stack begins at $00FF and counts down to $00C0.

Note: Using the stack area for data storage or temporary work locations requires care to prevent the data from being overwritten due to stacking from an interrupt or subroutine call.

4.3 ROM (MC68HC05BS8)

The user ROM consists of 10K-bytes of memory, from $1800 to $3FDF.

4.4 EPROM (MC68HC705BS8)

The user EPROM consists of 10K-bytes of memory, from $1800 to $3FDF.

4.5 EEPROM

The EEPROM consists of 512 bytes, from $0200 to $03FF. A charge pump is built on the chip for the operation of EEPROM. Programming and erasing are controlled by writing to the EEPROM Control register at address $0007.

4.5.1 EEPROM Control Register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
EEPCR	$0007				EEOSC	EER1	EER0	EELAT	EEPGM	0000 0000

EEOSC - EEPROM Charge Pump Oscillator Enable

1 (set) – Internal oscillator turned on to clock the EEPROM charge pump. It requires a time t_{RCON} to stabilize. (Min. 1µs).

0 (clear) – Internal oscillator turned off. EEPROM charge pump clocked by internal bus clock.

EER1, EER0 - EEPROM Erase Mode Select Bits

These two bits select one of the three erase modes. Refer to Table 4-1 below.

Table 4-1 Erase Mode Select

EER1	EER0	ERASE MODE
0	0	No erase
0	1	Byte erase
1	0	Block erase (block 1 or block 2)
1	1	Bulk erase (block 1 and block 2)

The EEPROM memory space is divided into two 256 byte blocks. Block 1 is located at address $0200-$02FF, and block 2 is located at $0300-$03FF. Providing the EELAT and EEPGM bits are “1”, the EER1 and EER0 bits indicate whether the access to the EEPROM is for an erase or programming purpose. Block protect function applies on block 2 of the EEPROM memory space.

EELAT - EEPROM Programming Latch Control

1 (set) – EEPROM address and data bus configured for programming (writes to EEPROM cause address and data to be latched). EEPROM is in programming mode and cannot be read when this bit is set.

0 (clear) – EEPROM address and data bus configured for normal reads. EER1, EER0, and EEPGM are forced to “0”s.

EEPGM - EEPROM Programming Power Enable

1 (set) – Programming power switched on to EEPROM array. If EELAT≠1 then EEPGM cannot be set.

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

4.5.2 EEPROM Options Register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
EEOPR	$0200							EEPRT	LVR	unaffected

EEPRT - EEPROM Protect

1 (set) – Block 2 ($300-$3FF) configured for read/write.

0 (clear) – Block 2 ($300-$3FF) configured for read only.

When this bit is erased to “1”, writing to block 2 is not possible until the next external or power-on reset occurs.

LVR - Low Voltage Reset

1 (set) – Low voltage reset function is disabled.

0 (clear) – Low voltage reset function is enabled.

This bit does not control any EEPROM operation, but enables/disables the LVR function. When enabled, and the MCU will reset if V_{DD} drops below V_{LVR}. When this bit is changed, its new value will have no effect until next external power-on reset.

4.5.3 Read Procedure

To read data from EEPROM, the EELAT bit must be cleared. EEPGM, EER1, and EER0 bits will be forced to zero. EEPROM is read as if it were normal ROM. The V_{PP} charge pump generator is off since EEPGM is zero. If a read is performed while ELAT is set, data will be read as $FF.

4.5.4 Erase Procedure

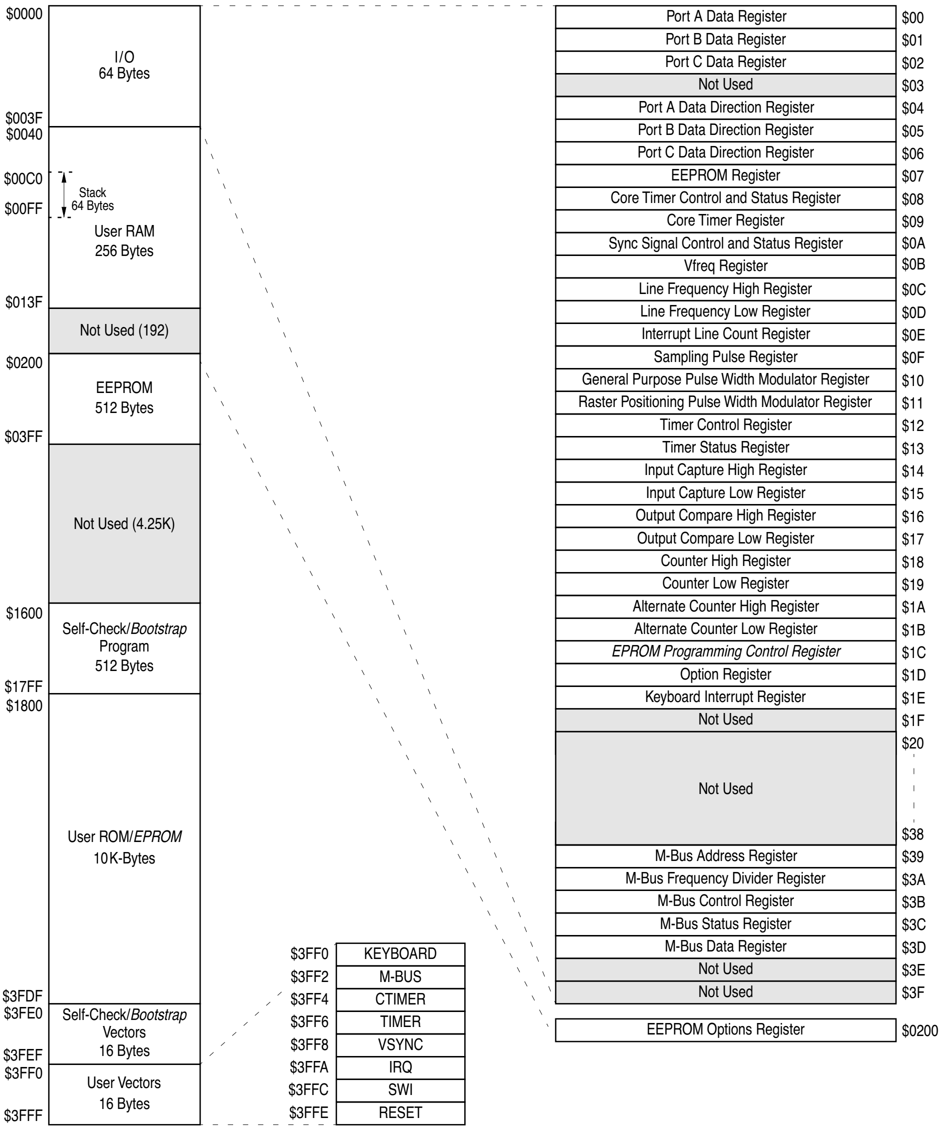
There are three types of ERASE operation mode (see Table 4-1): byte erase, block erase, and bulk erase.

1) To perform byte erase operation, set EELAT=1, EER1=0, and EER0=1, write any data to the address to be erase, and set EEPGM for a time t_{EBYTE}.
2) To perform block erase operation, set EELAT=1, EER1=1, and EER0=0, write any data to any address in the block, and set EEPGM for a time t_{EBLOCK}.
3) To perform bulk erase operation, set EELAT=1, EER1=1, and EER0=0, write any data to any address in the EEPROM map, and set EEPGM for a time t_{EBULK}.

Note: Erase operation to any part of block 2 is not possible if the EEPRT bit is programmed to “0”.

4.5.5 Programming Procedure

To program the content of EEPROM, set EELAT bits, write data to the desired address, and set the EEPGM bit. After the required programming delay t_{PROG}, EELAT must be cleared, which also resets EEPGM. During a programming operation, any access to the EEPROM will return $FF. To program a second byte, EELAT must be cleared before it is set, or the programming will have no effect.



Register	Address
Port A Data Register	$00
Port B Data Register	$01
Port C Data Register	$02
Not Used	$03
Port A Data Direction Register	$04
Port B Data Direction Register	$05
Port C Data Direction Register	$06
EEPROM Register	$07
Core Timer Control and Status Register	$08
Core Timer Register	$09
Sync Signal Control and Status Register	$0A
Vfreq Register	$0B
Line Frequency High Register	$0C
Line Frequency Low Register	$0D
Interrupt Line Count Register	$0E
Sampling Pulse Register	$0F
General Purpose Pulse Width Modulator Register	$10
Raster Positioning Pulse Width Modulator Register	$11
Timer Control Register	$12
Timer Status Register	$13
Input Capture High Register	$14
Input Capture Low Register	$15
Output Compare High Register	$16
Output Compare Low Register	$17
Counter High Register	$18
Counter Low Register	$19
Alternate Counter High Register	$1A
Alternate Counter Low Register	$1B
EPROM Programming Control Register	$1C
Option Register	$1D
Keyboard Interrupt Register	$1E
Not Used	$1F
Not Used	$20 $38
M-Bus Address Register	$39
M-Bus Frequency Divider Register	$3A
M-Bus Control Register	$3B
M-Bus Status Register	$3C
M-Bus Data Register	$3D
Not Used	$3E
Not Used	$3F
EEPROM Options Register	$0200

Address	Vector
$3FF0	KEYBOARD
$3FF2	M-BUS
$3FF4	CTIMER
$3FF6	TIMER
$3FF8	VSYNC
$3FFA	IRQ
$3FFC	SWI
$3FFE	RESET

Figure 4-1 Memory Map

Table 4-2 Register Outline

Register Name	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Port A data	$0000	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0	unaffected
Port B data	$0001	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0	unaffected
Port C data	$0002	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0	unaffected
Not used	$0003									
Port A data direction	$0004	DDRA7	DDRA6	DDRA5	DDRA4	DDRA3	DDRA2	DDRA1	DDRA0	0000 0000
Port B data direction	$0005	DDRB7	DDRB6	DDRB5	DDRB4	DDRB3	DDRB2	DDRB1	DDRB0	0000 0000
Port C data direction	$0006	DDRC7	DDRC6	DDRC5	DDRC4	DDRC3	DDRC2	DDRC1	DDRC0	0000 0000
EEPROM control	$0007				EEOSC	EER1	EER0	EELAT	EEPGM	0000 0000
Core timer control and status	$0008	CTOF	RTIF	CTOFE	RTIE			RT1	RT0	0000 0011
Core timer	$0009	CT7	CT6	CT5	CT4	CT3	CT2	CT1	CT0	0000 0000
Sync signal control and status	$000A	VPOL	HPOL	VDET	HDET	SOUT	INSRT	SIN1	SIN0	0000 0000
Vfreq	$000B	VF7	VF6	VF5	VF4	VF3	VF2	VF1	VF0	0000 0000
Line frequency high	$000C	VF8				LF11	LF10	LF9	LF8	0000 0000
Line frequency low	$000D	LF7	LF6	LF5	LF4	LF3	LF2	LF1	LF0	0000 0000
Interrupt line counter	$000E	VSIE	LC6	LC5	LC4	LC3	LC2	LC1	LC0	0000 0010
Sampling pulse	$000F		SP6	SP5	SP4	SP3	SP2	SP1	SP0	0000 0010
General PWM	$0010	ODE	CRE	GPW5	GPW4	GPW3	GPW2	GPW1	GPW0	0000 0000
Raster positioning PWM	$0011	RSP	RSPW6	RSPW5	RSPW4	RSPW3	RSPW2	RSPW1	RSPW0	0000 0000
Timer control	$0012	ICIE	OCIE	TOIE				IEDG	OLVL	0000 00u1
Timer status	$0013	ICF	OCF	TOF						uuu0 0000
Input capture high	$0014	IC15	IC14	IC13	IC12	IC11	IC10	IC9	IC8	unaffected
Input capture low	$0015	IC7	IC6	IC5	IC4	IC3	IC2	IC1	IC0	unaffected
Output compare high	$0016	OC15	OC14	OC13	OC12	OC11	OC10	OC9	OC8	unaffected
Output compare low	$0017	OC7	OC6	OC5	OC4	OC3	OC2	OC1	OC0	unaffected
Counter high	$0018	TC15	TC14	TC13	TC12	TC11	TC10	TC9	TC8	$FF
Counter low	$0019	TC7	TC6	TC5	TC4	TC3	TC2	TC1	TC0	$FC
Alternate counter high	$001A	AC15	AC14	AC13	AC12	AC11	AC10	AC9	AC8	$FF
Alternate counter low	$001B	AC7	AC6	CA5	AC4	AC3	AC2	AC1	AC0	$FC
EPROM programming control	*$001C*	*Reserved*						*ELAT*	*PGM*	*0000 0000*
Option	$001D	INTO	COP							0100 0000
Keyboard interrupt	$001E		KBIC	KBE5	KBE4	KBE3	KBE2	KBE1	KBE0	0000 0000
Not used	$001F									

Table 4-2 Register Outline

Register Name	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Not used	$0020 to $0038									
M-Bus address	$0039	MAD7	MAD6	MAD5	MAD4	MAD3	MAD2	MAD1		0000 000-
M-Bus frequency divider	$003A				FD4	FD3	FD2	FD1	FD0	0000 0000
M-Bus control	$003B	MEN	MIEN	MSTA	MTX	TXAK		SIFC	SIIC	0000 0000
M-Bus status	$003C	MCF	MASS	MBB	MAL	SIF	SRW	MIF	RXAK	1000 0001
M-Bus data	$003D	MD7	MD6	MD5	MD4	MD3	MD2	MD1	MD0	undefined
Not used	$003E									
Not used	$003F									

EEPROM options	$0200							EEPRT	LVR	unaffected

THIS PAGE LEFT BLANK INTENTIONALLY

5
RESETS AND INTERRUPTS

The section describes the reset and interrupt functions available on the MC68HC05BS8.

5.1 RESETS

The MC68HC05BS8 can be reset in four ways:

- by the initial power-on reset function, (POR)
- by an active low input to the $\overline{\text{RESET}}$ pin, ($\overline{\text{RESET}}$)
- by a COP watchdog timer reset, (COPR)
- and by a Low Voltage Reset, (LVR)

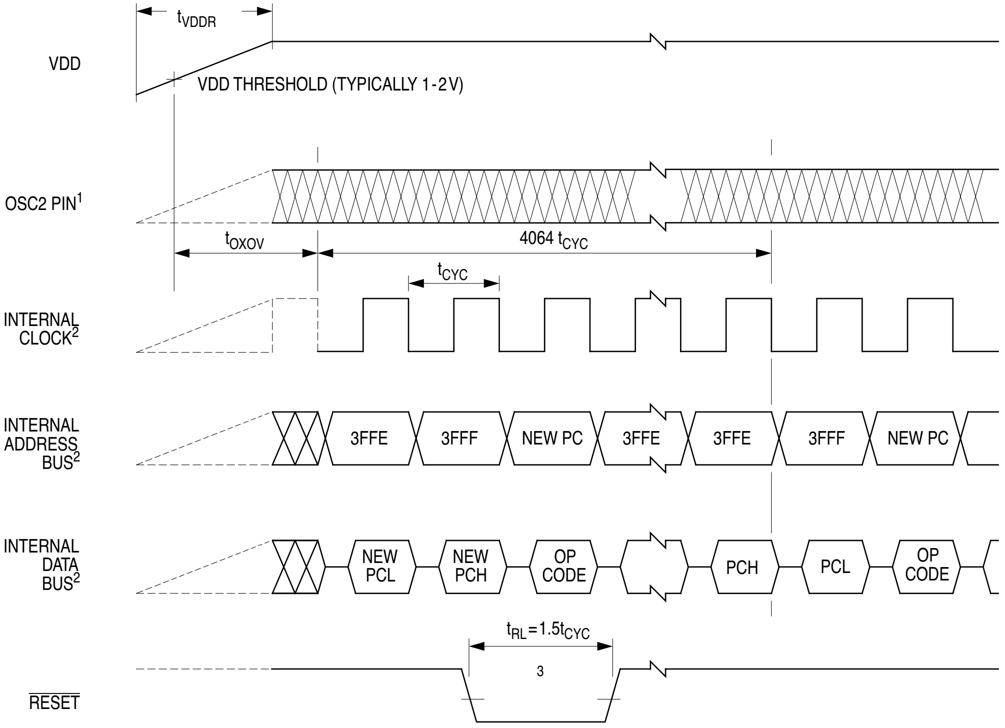
Any of these resets will cause the program to go to its starting address, specified by the contents of memory locations $3FFE and $3FFF, and cause the interrupt mask (I-bit) of the Condition Code register to be set.

5.1.1 Power-On Reset (POR)

The power-on reset occurs when a positive transition is detected on the supply voltage, V_{DD}. The power-on reset is used strictly for power-up conditions, and should not be used to detect any drops in the power supply voltage. There is no provision for a power-down reset. The power-on circuitry provides for a 4064 t_{CYC} delay from the time that the oscillator becomes active. If the external $\overline{\text{RESET}}$ pin is low at the end of the 4064 t_{CYC} 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.

5.1.2 $\overline{\text{RESET}}$ 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 $1.5t_{CYC}$. The $\overline{\text{RESET}}$ pin contains an internal Schmitt Trigger as part of its input to improve noise immunity.



NOTES:
1. OSC2 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 5-1 Power-On Reset and $\overline{\text{RESET}}$ Timing

5.1.3 Low Voltage Reset (LVR)

When the LVR function is enabled, an internal reset is generated if V_{DD} drops below V_{LVR}. (See Section 13 for value of V_{LVR}.)

This LVR reset function is enabled/disabled by programming or erasing bit 0 in the EEPROM Options register ($0200). Refer to Section 4.5.2.

5.1.4 Computer Operating Properly (COP) Reset

The MC68HC05BS8 contains a watchdog timer that automatically times out if this timer is not reset (cleared) within a specific amount of time by a program reset sequence.

Note: COP time-out is prevented by periodically writing a “0” to bit 0 of address $3FF0.

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

The COP reset function is enabled after a reset, and it can be disabled by writing a “0” to bit 6 in the Option register at address $001D. Once disabled, it cannot be enabled except by a reset function.

See Section 6.2.3 for more information on the COP watchdog timer.

5.2 INTERRUPTS

The MC68HC05BS8 can be interrupted by different sources – six maskable hardware interrupt and one non-maskable software interrupt:

- Software Interrupt Instruction (SWI)
- External signal on the $\overline{\text{IRQ}}$ pin
- Sync Signal Processor (SSP)
- Programmable Timer (TIMER)
- Core Timer (CTIMER)
- M-Bus Interface (MBUS)
- Keyboard (KBI)

If the interrupt mask bit (I-bit) of the CCR is set, all maskable interrupts (internal and external) are disabled. Clearing the I-bit enables interrupts.

Interrupts cause the processor to save the register contents on the stack and to set the interrupt mask (I-bit) to prevent additional interrupts. The RTI instruction causes the register contents to be recovered from the stack and normal processing to resume.

Unlike reset, hardware interrupts do not cause the current instruction execution to be halted, but are considered pending until the current instruction is complete. The current instruction is the one already fetched and being operated on. When the current instruction is complete, the processor checks all pending hardware interrupts. If interrupts are not masked (CCR I-bit clear) the processor proceeds with interrupt processing; otherwise, the next instruction is fetched and executed. Table 5-1 shows the relative priority of all the possible interrupt sources.

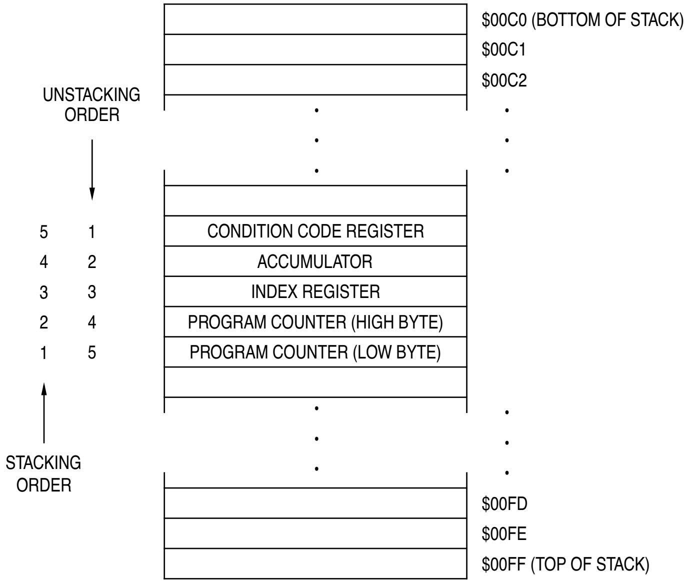


Figure 5-2 Interrupt Stacking Order

Table 5-1 Reset/Interrupt Vector Addresses

Register	Flag Name	Interrupt	CPU Interrupt	Vector Address	Priority
–	–	Reset	$\overline{\text{RESET}}$	$3FFE-$3FFF	highest
–	–	Software	SWI	$3FFC-$3FFD	↑
–	–	External Interrupt	$\overline{\text{IRQ}}$	$3FFA-$3FFB	
SSCR	–	VSYNC	SSP	$3FF8-$3FF9	
TSR	TOF	Timer Overflow	TIMER	$3FF6-$3FF7	
	OCF	Output Compare			
	ICF	Input Capture			
CTCSR	CTOF	Core Timer Overflow	CTIMER	$3FF4-$3FF5	
	RTIF	Core Timer Interrupt			
MSR	MIF	M-Bus	MBUS	$3FF2-$3FF3	
	SIF				
–	–	Keyboard	KBI	$3FF0-$3FF1	lowest

5.2.1 Non-maskable Software Interrupt (SWI)

The software interrupt (SWI) is an executable instruction and a non-maskable interrupt: it is execute regardless of the state of the I-bit in the CCR. If the I-bit is zero (interrupt enabled), SWI is executed after interrupts that were pending when the SWI was fetched, but before interrupts generated after the SWI was fetched. The SWI interrupt service routine address is specified by the contents of memory locations $3FFC and $3FFD.

5.2.2 Maskable Hardware Interrupts

If the interrupt mask bit (I-bit) of the CCR is set, all maskable interrupts (internal and external) are masked. Clearing the I-bit allows interrupt processing to occur.

Note: The internal interrupt latch is cleared in the first part of the interrupt service routine; therefore, one external interrupt pulse could be latched and serviced as soon as the I-bit is cleared.

5.2.2.1 External Interrupt ($\overline{\text{IRQ}}$)

The external interrupt $\overline{\text{IRQ}}$ can be software configured for “negative-edge” or “negative-level” sensitive triggering by the INTO bit in the Option register.

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Option register	$001D	INTO	COP							01-- ----

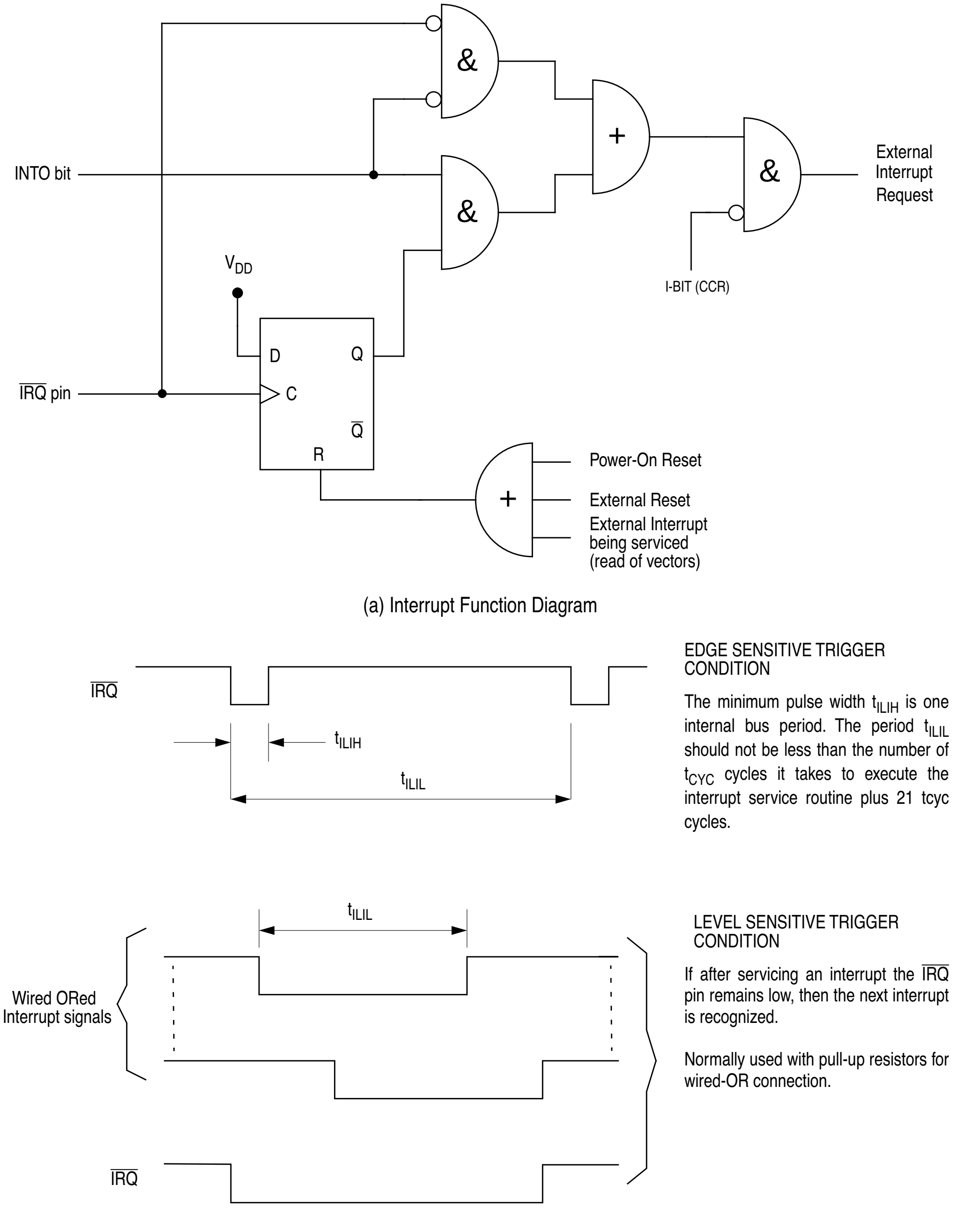
INTO

1 (set) – Negative-edge sensitive triggering for $\overline{\text{IRQ}}$.

0 (clear) – Negative-level sensitive triggering for $\overline{\text{IRQ}}$.

When the signal of the external interrupt pin, $\overline{\text{IRQ}}$, satisfies the condition selected, 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 $3FFA & $3FFB.

The interrupt logic recognizes negative edge transitions and pulses (special case of negative edges) on the external interrupt line. Figure 5-3 shows both a block diagram and timing for the interrupt line ($\overline{\text{IRQ}}$) to the processor. The first method is used if pulses on the interrupt line are spaced far enough apart to be serviced. The minimum time between pulses is equal to the number of cycles required to execute the interrupt service routine plus 21 cycles. Once a pulse occurs, the



(a) Interrupt Function Diagram

EDGE SENSITIVE TRIGGER CONDITION

The minimum pulse width t_{ILIH} is one internal bus period. The period t_{ILIL} should not be less than the number of t_{CYC} cycles it takes to execute the interrupt service routine plus 21 tcyc cycles.

LEVEL SENSITIVE TRIGGER CONDITION

If after servicing an interrupt the $\overline{\text{IRQ}}$ pin remains low, then the next interrupt is recognized.

Normally used with pull-up resistors for wired-OR connection.

(b) Interrupt Mode Diagram

Figure 5-3 External Interrupt Circuit and Timing

next pulse should not occur until the MCU software has exited the routine (an RTI occurs). The second configuration shows several interrupt lines wired-OR to perform the interrupt at the processor. Thus, if the interrupt lines remain low after servicing one interrupt, the next interrupt is recognized.

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

5.2.2.2 Sync Signal Processor Interrupt

The CPU will process an Sync Signal Processor VSYNC interrupt if the following conditions are satisfied:

1) the I-bit of the CCR is cleared,

2) the VSIE bit of the Interrupt Line Count register (ILCR) is set, and

3) the value of the horizontal line counter matches the value set in the ILCR.

This interrupt will vector to the interrupt service routine located at the address specified by the contents of $3FF8 and $3FF9. The VSYNC interrupt latch will be cleared automatically by fetching of these vectors.

Refer to Section 9 for detailed description of Sync Signal Processor.

5.2.2.3 M-Bus Interrupts

The hardware M-Bus interrupt is enabled when the M-Bus Interrupt Enable bit (MIEN) of M-Bus Control register is set, provided the interrupt mask bit of the Condition Code register is cleared. The interrupt service routine address is specified by the contents of memory location $3FF2 and $3FF3.

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
M-Bus Status Register	$001A	MCF	MAAS	MBB	MAL	SIF	SRW	MIF	RXAK	1000 0001

MIF - M-Bus Interrupt

1 (set) – An M-Bus interrupt has occurred.

0 (clear) – An M-Bus interrupt has not occurred.

When this bit is set, an interrupt is generated to the CPU if MIEN is set. This bit is set when one of the following events occurs:

1) Completion of one byte of data transfer. It is set at the falling edge of the 9th clock - MCF set.

2) A match of the calling address with its own specific address in slave mode - MAAS set.

3) A loss of bus arbitration - MAL set.

This bit must be cleared by software in the interrupt routine.

MCF - Data Transfer Complete

1 (set) – A byte transfer has been completed.

0 (clear) – A byte is being transfer.

MAAS - Addressed as Slave

1 (set) – Currently addressed as a slave.

0 (clear) – Not currently addressed.

Then CPU needs to check the SRW bit and set its MTX bit accordingly. Writing to the M-Bus Control register clears this bit.

MAL - Arbitration Lost

1 (set) – Lost arbitration in master mode.

0 (clear) – No arbitration lost.

SIF - Software M-Bus Interrupt

1 (set) – Signals on PC6 and PC7 pins satisfy the "start" condition for the "soft" M-Bus protocol. This bit cannot be set if the SIIC bit in the MCR is cleared. An interrupt to the CPU is generated only if the I-bit in the CCR is also cleared.

0 (clear) – No "soft" M-Bus interrupt.

Refer to Section 8 for detailed description of M-Bus Interface.

5.2.2.4 Timer Interrupts

There are three interrupt sources from the 16-bit free-running counter timer.

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Timer Status Register	$0013	ICF	OCF	TOF						uuu0 0000

ICF - Input Capture Flag

This bit is set when a proper edge has been sensed by the input capture edge detector. It is cleared by reading the TSR (with ICF set) followed by accessing the Input Capture register LSB ($0015).

OCF - Output Compare Flag

This bit 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 LSB ($0017).

TOF - Timer Overflow Flag

This bit is set during the counter transition from $FFFF to $0000. It is cleared by reading the TSR (with TOF set) followed by reading the counter LSB ($0019).

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 $FFF6 and $FFF7.

Refer to section 6.1 for detailed description of the 16-bit Counter Timer.

5.2.2.5 Core Timer Interrupts

There are two interrupt sources, TOF and RTIF bits of Multi-Function Timer Control and Status Register. The interrupt service routine address is specified by the contents of memory location $3FF4 and $3FF5.

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
CTimer Control and Status Register	$0008	CTOF	RTIF	CTOFE	RTIE			RT1	RT0	0000 0011

CTOF - Timer Overflow

1 (set) – CTimer counter overflow has occurred.

0 (clear) – No CTimer counter overflow has occurred.

This bit is set when the 8-bit ripple counter overflows from $FF to $00; a timer overflow interrupt will occur, if CTOFE is set. CTOF is cleared by writing a “0” to the bit.

RTIF - Real Time Interrupt Flag

1 (set)	–	A real time interrupt has occurred.
0 (clear)	–	A real time interrupt has not occurred.

Refer to Section 6.2 for detailed description of the CTimer.

5.2.2.6 Keyboard Interrupt

Keyboard interrupt functions are available on PC0-PC5. Each port pin can be individually configured for the keyboard interrupt function in the KBI register at $0010. Once configured, an interrupt is recognized by a high to low transition (negative edge) sensed on the pin and the I-bit in the CCR is also cleared.

The interrupt service routine is specified by the contents of the memory locations $3FF0 and $3FF1.

6
TIMERS

6.1 PROGRAMMABLE TIMER

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

Because the timer has a 16-bit architecture, the I/O registers for the input capture and output compare functions are pairs of 8-bit registers (high byte and low byte). Generally, assessing the low byte of a specific timer function allows full control of that function. However, an access of the high byte inhibits that specific timer function until the low byte is also accessed.

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

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

–	Timer Control Register (TCR)	$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.

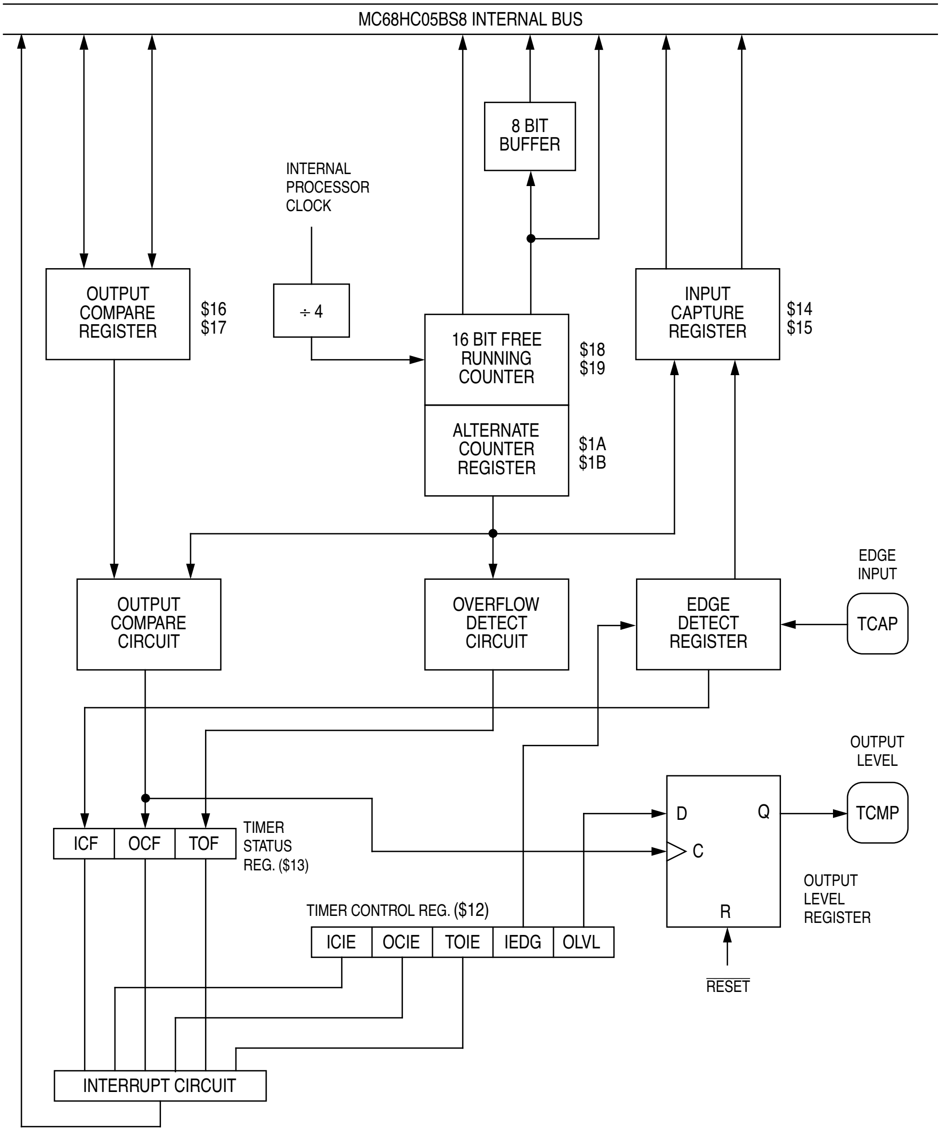


Figure 6-1 Programmable Timer Block Diagram

6.1.1 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 0.95µs if the internal bus clock is 4.2MHz. 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 and $19 (counter register) or $1A and $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 TCR) 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.

6.1.2 Output Compare Register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
OCMPH	$0016	OC15	OC14	OC13	OC12	OC11	OC10	OC9	OC8	unaffected
OCMPL	$0017	OC7	OC6	OC5	OC4	OC3	OC2	OC1	OC0	unaffected

The 16-bit 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. The Output Compare register’ value 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 register 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 initialize clearing of OCF;
3) write to Output Compare register Low-byte to enable the output compare function.

6.1.3 Input Capture Registers

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
ICAPH	$0014	IC15	IC14	IC13	IC12	IC11	IC10	IC9	IC8	unaffected
ICAPL	$0015	IC7	IC6	IC5	IC4	IC3	IC2	IC1	IC0	unaffected

‘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.

6.1.4 Timer Control Register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
TCR	$0012	ICIE	OCIE	TOIE				IEDG	OLVL	0000 00u1

The TCR is a read/write register containing five control bits. Four bits control interrupts associated with each of the four flag bits found in the Timer Status register. The other bit controls which edge is significant to the input capture edge detector. The Timer Control register and the free-running counter are the only sections of the timer affected by reset.

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.

When OLVL is set high output level will be clocked into the output level register by the next successful output compare on the TCMP pin.

6.1.5 Timer Status Register (TSR)

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
TSR	$0013	ICF	OCF	TOF	0	0	0	0	0	uuu0 0000

The Timer Status register contains the status bits for the above three interrupt conditions - ICF, OCF, and 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 the 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.

6.1.6 Programmable Timer Timing Diagrams

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

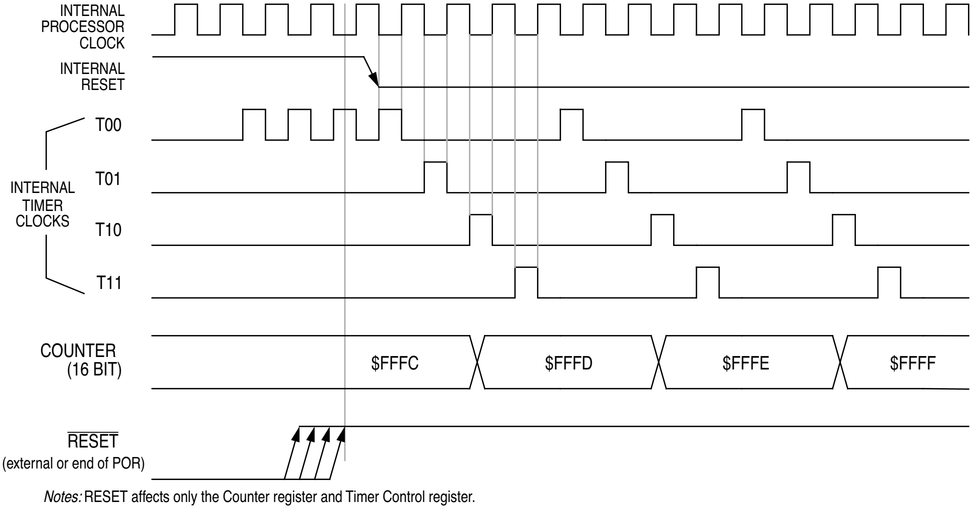


Figure 6-2 Timer State Timing Diagram for Reset

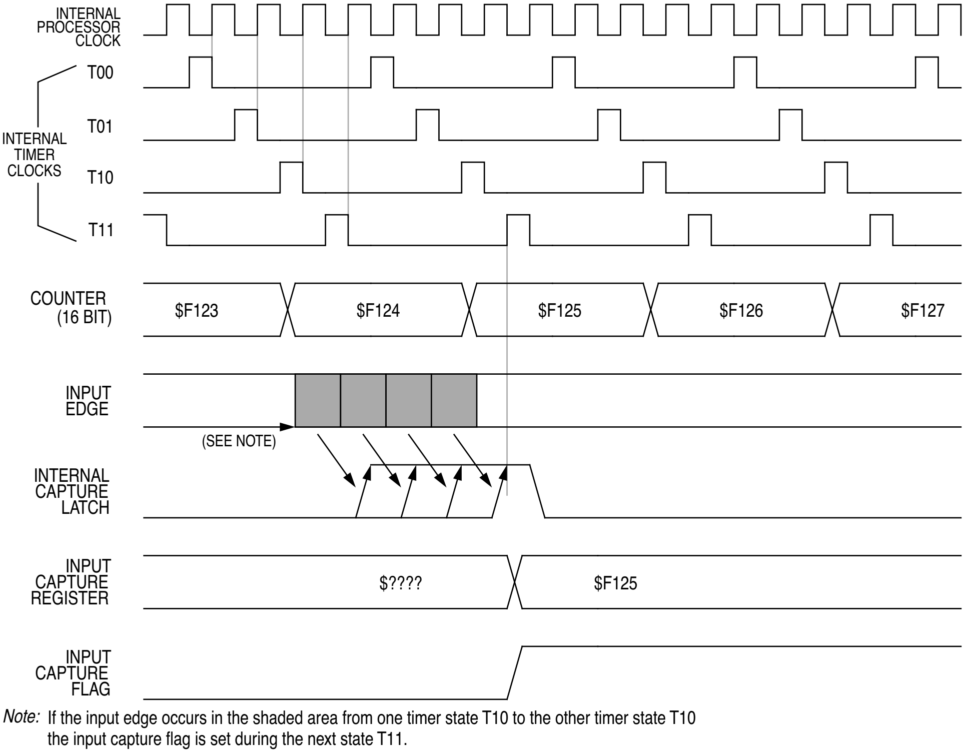
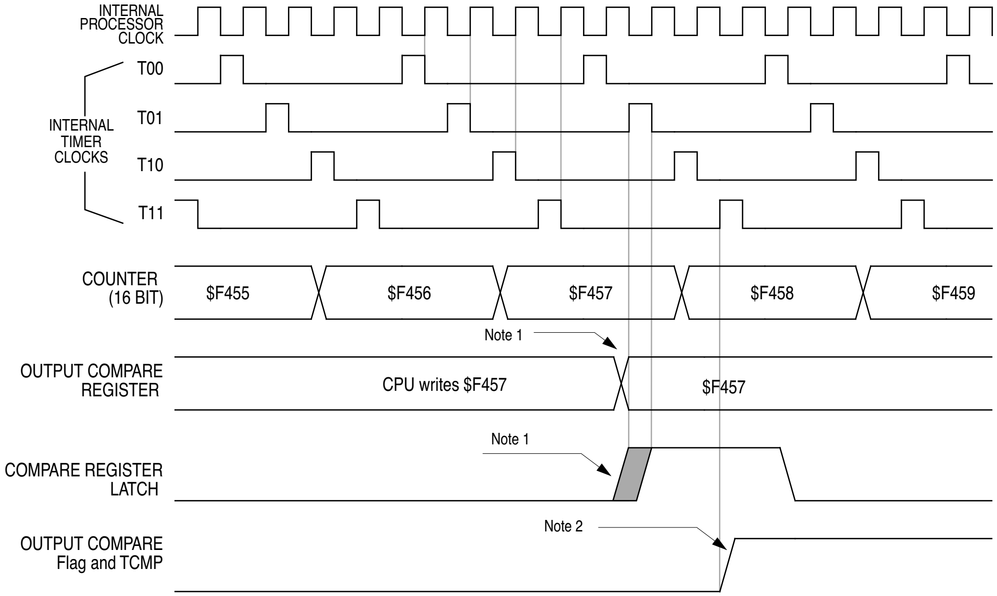


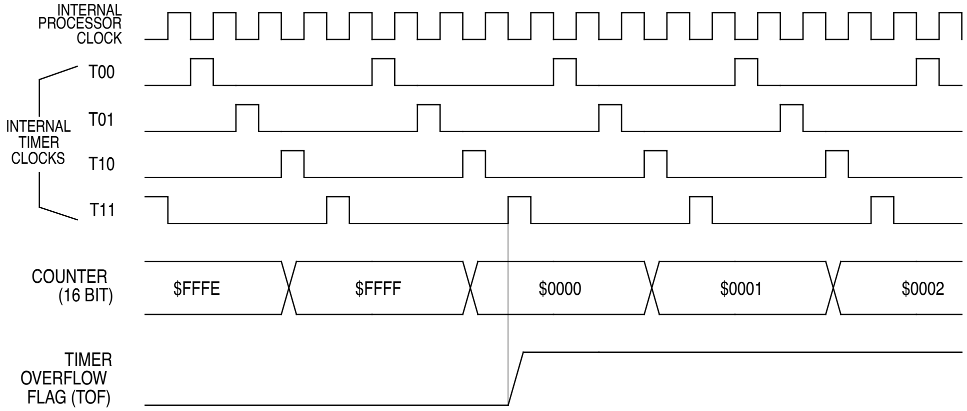
Figure 6-3 Timer State Timing Diagram for Input Capture



Note: 1. The CPU write to the compare registers may take place at any time, but a compare only occurs at the timer state T01. Thus a 4-cycle difference may exist between the write to the compare register and the actual compare.

2. The output compare flag is set at the timer state T11 that follows the comparison match ($F547 in this example).

Figure 6-4 Timer State Timing Diagram for Output Compare



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

Figure 6-5 Timer State Diagram for Timer Overflow

6.2 CORE TIMER

The Core Timer is a 15-stage multi-functional ripple counter which provides miscellaneous function to the MC68HC05BS8 MCU. It includes a timer overflow function, real-time interrupt, and COP watchdog.

As seen in Figure 6-6, the Timer is driven by the internal bus clock divided by four with a fixed prescaler. This signal drives an 8-bit ripple counter. The value of this 8-bit ripple counter can be read by the CPU at any time by accessing the Ctimer Counter register (CTCR) at address $09. A timer overflow function is implemented on the last stage of this counter, giving a possible interrupt at the rate of E/1024. Four additional stages produces a resulting clock of E/16384, driving the Real Time Interrupt circuit. The RTI circuit consists of three divider stages with a 1 of 4 selector. The output of the RTI circuit is further divided by eight to drive the optional COP Watchdog Timer circuit. The RTI rate selector bits, and the RTI and CTOF enable bits and flags are located in the CTimer Control and Status register (CTCSR) at location $08.

6.2.1 CTimer Counter Register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
CTCR	$0009	CT7	CT6	CT5	CT4	CT3	CT2	CT1	CT0	0000 0000

The Core Timer Counter register is a read-only register which contains the current value of the 8-bit ripple counter. This counter is clocked at f_{OP}/4 and can be used for various functions including a software capture. Extended time periods can be attained using the TOF function to increment a temporary RAM storage location thereby simulating a 16-bit (or more) counter.

During the Power-on reset (POR) cycle, all CTimer counters are first cleared, the counters then count 4064 cycles before it is cleared again. After this 4064 cycles, the POR circuit releases the device from reset. At this point, if $\overline{\text{RESET}}$ is not asserted, the timer will start counting up from zero and normal device operation will begin. If $\overline{\text{RESET}}$ is asserted anytime during operation (other than POR), the counter chain will be cleared.

6.2.2 CTimer Control and Status Register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
CTCSR	$0008	CTOF	RTIF	CTOFE	RTIE	0	0	RT1	RT0	0000 0011

CTOF - CTimer Overflow

1 (set) – 8-bit ripple timer overflow has occurred.

0 (clear) – No 8-bit ripple timer overflow has occurred.

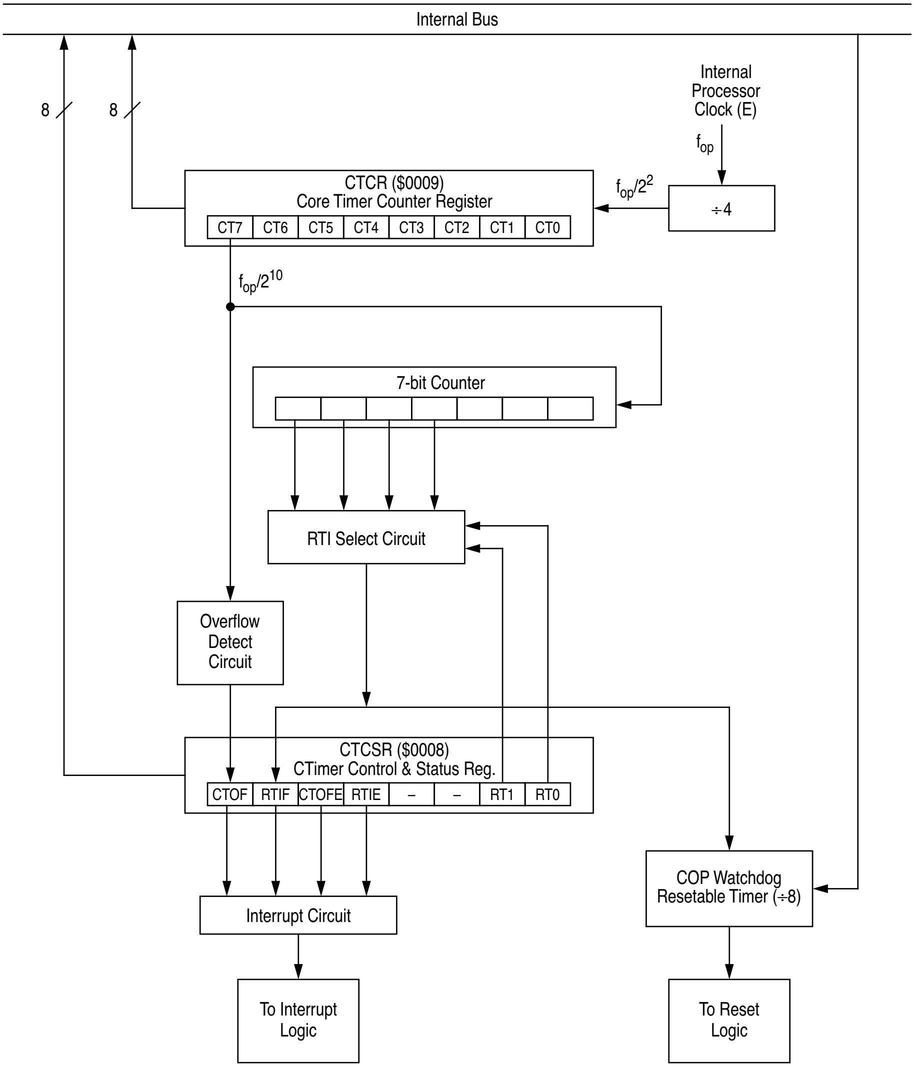


Figure 6-6 Core Timer Block Diagram

This bit is set when the 8-bit ripple counter overflows from $FF to $00; a timer overflow interrupt will occur, if TOFE (bit 5) is set. TOF is cleared by writing a ‘0’ to the bit.

RTIF - Real Time Interrupt Flag

1 (set) – A real time interrupt has occurred.

0 (clear) – A real time interrupt has not occurred.

When RTIF is set, a CPU interrupt request is generated if RTIE is set. The clock frequency that drives the RTI circuit is E/12^{13}. The RTI rate is selectable by the RT1 and RT0 bits. RTIF is cleared by writing a “0” to the bit.

CTOFE - CTimer Overflow Interrupt Enable

1 (set) – TOF interrupt is enabled.

0 (clear) – TOF interrupt is disabled.

RTIE - Real Time Interrupt Enable

1 (set) – Real time interrupt is enabled.

0 (clear) – Real time interrupt is disabled.

RT1, RT0 - Rate Select for COP watchdog and RTI

See Section 6.2.3 on COP watchdog reset.

6.2.3 COP Watchdog Reset

The COP (Computer Operating Properly) watchdog timer function is implemented by using the output of the RTI selected output. The minimum COP reset rates are determined by RT0 and RT1 of CTimer Control and Status register. 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 periodically writing a “0” to bit 0 of address $3FF0.

The COP reset function is enabled after a reset, and can be disabled by writing a “0” to the COP bit (bit 6) in the Option Register at address $001D. Once disabled, it cannot be enabled except by a reset function. Also, STOP mode cannot be entered when the COP watchdog is enabled.

Table 6-1 COP Reset and RTI Rates

RT1	RT0	Bus Frequency, f_{OP}=2.1 MHz		
		Divide Ratio	Minimum RTI Rate	Minimum COP Reset Rate (RTI x 7) 1/(f_{OP}/Divide Ratio/7)
0	0	2^{14}	7.81 ms	54.7 ms
0	1	2^{15}	15.6 ms	109 ms
1	0	2^{16}	31.2ms	219 ms
1	1	2^{17}	62.5 ms	438 ms
RT0 and RT1 should only be changed immediately after COP watchdog timer has been reset.				

7

PULSE WIDTH MODULATION

The MC68HC05BS8 has two independent PWMs; one general purpose (GPWM) and one raster positioning (RSPWM). They are controlled by the PWM registers located at $0010 and $0011.

7.1 General Purpose Pulse Width Modulator

The GPWM consists of a comparator and a 6-bit free-running counter driven by the internal bus clock. The counter runs from $00 to $3E and rolls over back to $00. Whenever the GPWM value in the GPWM register is greater than the running counter, the output of the GPWM pin will be "high" (See Figure 7-1).

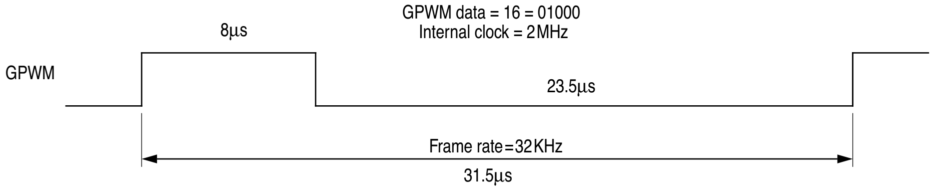


Figure 7-1 GPWM Timing Example

To reduce the occurrence of fast logic switching edges on the PCB, the GPWM output is connected to the output pin via an on chip resistor of nominal value of 5KΩ. In this way, for use with DC controls the RC filter capacitor is connected externally directly to the output. A further advantage is that the PWM logic signal has only to overcome modest on-chip capacitances and thus supply current spikes are significantly reduced.

The GPWM output can be configured as an open drain output by setting ODE bit in the GPWM register. It should be noted that the diode associated with the P-channel device is still connected to the output pin and therefore the normal voltage limitations apply, i.e. up to V_{DD}. Figure 7-2 shows a schematic of the GPWM output.

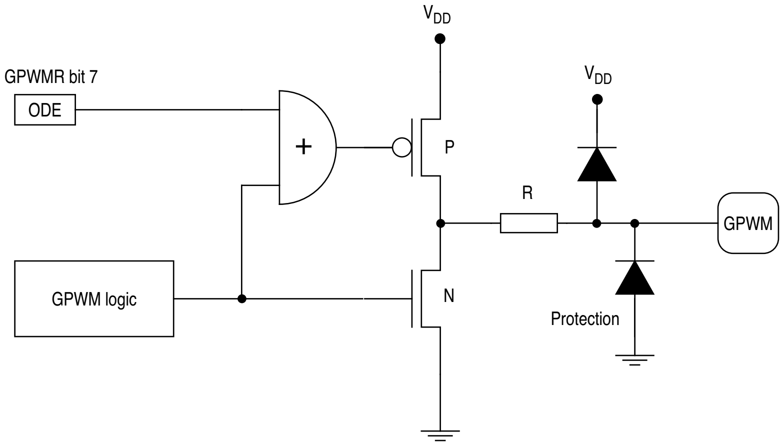


Figure 7-2 GPWM Output Configuration

7.1.1 General Purpose Pulse Width Modulator Register (GPWM)

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
GPWMR	$0010	ODE	CRE	GPW5	GPW4	GPW3	GPW2	GPW1	GPW0	0000 0000

ODE - Open Drain Enable

1 (set) – GPWM output configured as open drain output.

0 (clear) – GPWM output configured as direct drive output.

GPW5:0 - GPW Data

These six bits contain the pulse width modulator data for the GPWM.

7.2 Raster Positioning Pulse Width Modulator

The RSPWM consists of a comparator and a 7-bit counter clocked by the positive edge of a clock from the RCLK pin, and reset by the an active high signal from the PB7 pin. The RSPWM output will be low after a counter reset, and high when the counter value matches the value in the RSPWM register. Figure 7-3 shows example timings.

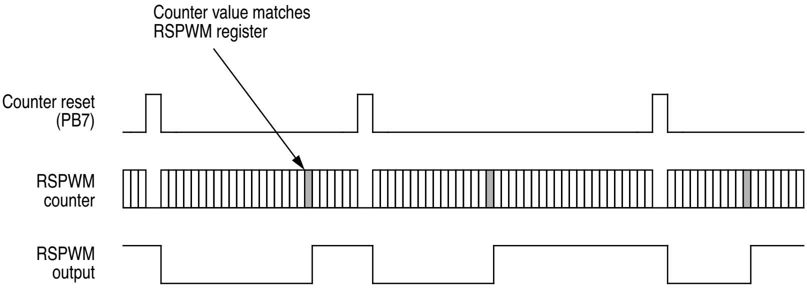


Figure 7-3 RSPWM Timing Example

RSA and RSB are the output pins of the RSPWM. They are connected such that, one of them is the PWM output waveform, while the other is always zero. This is controlled by the Raster Polarity bit (RSP) in the RSPWM register.

PB7 must be configured as an input port pin and the CRE bit in GPWMR must be set before it can be used as the input pin for the RSPWM reset signal. Figure 7-4 shows a block diagram of the RSPWM.

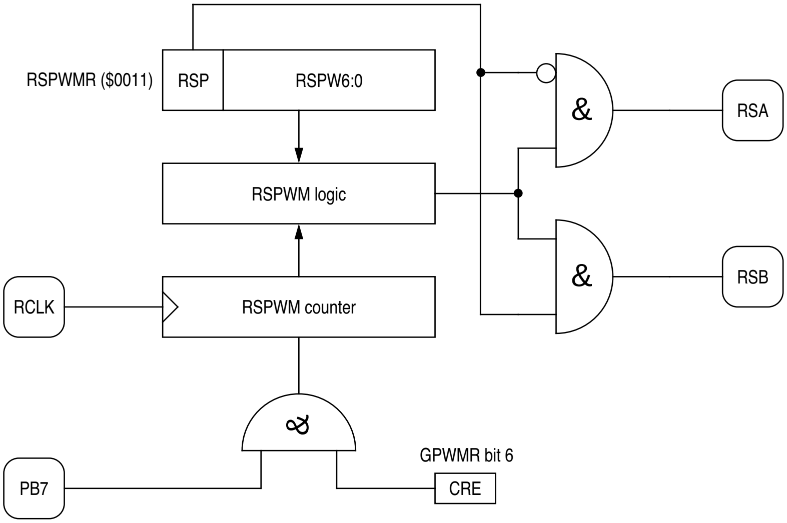


Figure 7-4 RSPWM Block Diagram

When used for raster position control, the RCLK is a clock signal which is in synchronized and with a frequency of multiples to the horizontal sync signal. The counter reset signal to PB7 pin is the horizontal fly back signal. Both of these signals will normally come from an OSD circuit.

7.2.1 Raster Positioning Pulse Width Modulator Register (RSPWM)

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
RSPWMR	$0011	RSP	RSPW6	RSPW5	RSPW4	RSPW3	RSPW2	RSPW1	RSPW0	0000 0000

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
GPWMR	$0010	ODE	CRE	GPW5	GPW4	GPW3	GPW2	GPW1	GPW0	0000 0000

RSP - Raster Polarity

1 (set) – RSA output is zero. RSPWM waveform is output on RSB.

0 (clear) – RSB output is zero. RSPWM waveform is output on RSA.

RSPW6:0 - RSPWM Data

These 7 bits contain the pulse width modulator data for the RSPWM. The RSPWM output remains low after the counter reset, and turns high when the counter value matches this value.

CRE - Clear Reset Enable

1 (set) – PB7 pin configured as reset input to the RSPWM. The DDR for PB7 should also be “0”.

0 (clear) – PB7 pin normal I/O operation.

8
M-BUS SERIAL INTERFACE

The MC68HC05BS8 MCU has two M-Bus Serial Interface; one is full hardware, and the other is software supported. Section 8.1 to Section 8.4 details the full hardware M-Bus interface. Section 8.5 details the software supported M-Bus.

M-Bus (Motorola Bus) is a two-wire, bidirectional serial bus which provides a simple, efficient way for data exchange between devices. It is fully compatible with the I^{2}C bus standard. This two-wire bus minimizes the interconnection between devices and eliminates the need for address decoders; resulting in less PCB traces and economic hardware structure. This bus is suitable for applications requiring communications in a short distance among a number of devices. The maximum data rate is 100 Kbit/s. The maximum communication length and number of devices that can be connected are limited by a maximum bus capacitance of 400 pF.

The M-Bus system is a true multi-master bus, including arbitration to prevent data collision if two or more masters intend to control the bus simultaneously. It may be used for rapid testing and alignment of end products via external connections to an assembly-line computer.

8.1 M-Bus Interface Features

- Compatible with I^{2}C bus standard
- Multi-master operation
- 32 software programmable serial clock frequencies
- Software selectable acknowledge bit
- Interrupt driven byte-by-byte data transfer
- Arbitration lost driven interrupt with automatic mode switching from master to slave
- Calling address identification interrupt
- Generate/detect the start, stop and acknowledge signals
- Repeated START signal generation
- Bus busy detection

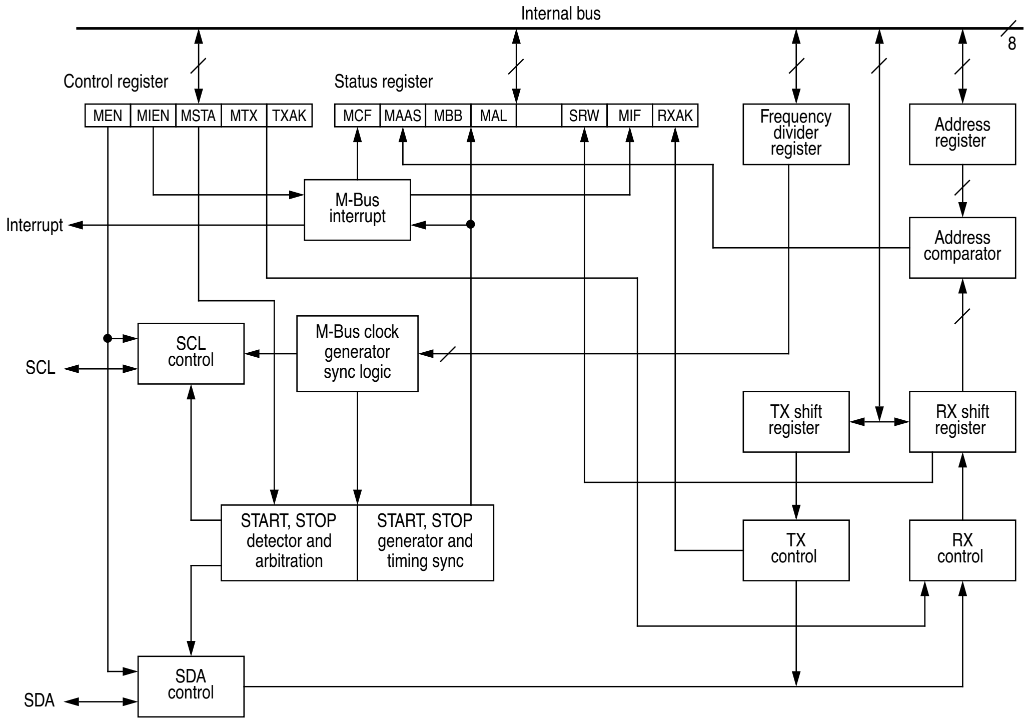


Figure 8-1 M-Bus Interface Block Diagram

8.2 M-Bus Protocol

Normally, a standard communication is composed of four parts,

1) START signal,
2) slave address transmission,
3) data transfer, and
4) STOP signal.

They are described briefly in the following sections and illustrated in Figure 8-2.

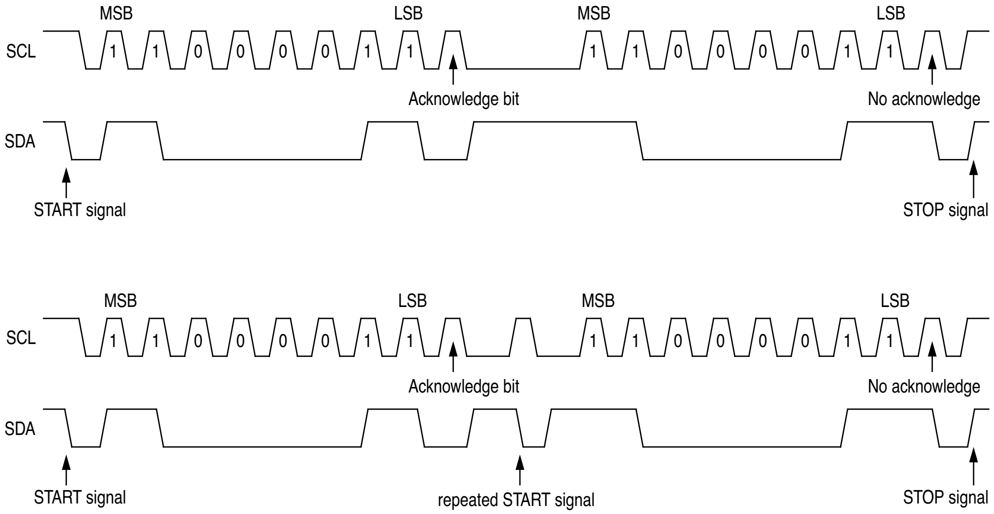


Figure 8-2 M-Bus Transmission Signal Diagram

8.2.1 START Signal

When the bus is free, i.e., no master device is occupying the bus (both SCL and SDA lines are at logic high), a master may initiate communication by sending a START signal. As shown in Figure 8-2, a START signal is defined as a high to low transition of SDA while SCL is high. This signal denotes the beginning of a new data transfer (each data transfer may contain several bytes of data) and wakes up all slaves.

8.2.2 Slave Address Transmission

The first byte of data transfer immediately following the START signal is the slave address transmitted by the master. This is a seven bits long calling address followed by a R/$\overline{W}$ bit. The R/$\overline{W}$ bit dictates the slave of the desired direction of data transfer.

Only the slave with matched address will respond by sending back an acknowledge bit by pulling the SDA low at the 9th clock; see Figure 8-2.

8.2.3 Data Transfer

Once a successful slave addressing is achieved, the data transfer can proceed byte by byte in a direction specified by the R/$\overline{W}$ bit sent by the calling master.

Each data byte is 8 bits long. Data can be changed only when SCL is low and must be held stable when SCL is high as shown in Figure 8-2. One clock pulse is for one bit of data transfer, MSB is transferred first. Each data byte has to be followed by an acknowledge bit. Hence, one complete data byte transfer requires 9 clock pulses.

If the slave receiver does not acknowledge the master, the SDA line should be left high by the slave, the master can then generate a STOP signal to abort the data transfer or a START signal (repeated START) to commence a new calling.

If the master receiver does not acknowledge the slave transmitter after one byte transmission, it means an “end of data” to the slave. The slave shall release the SDA line for the master to generate STOP or START signal.

8.2.4 Repeated START Signal

As shown in Figure 8-2, a repeated START signal is to generate a START signal without first generating a STOP signal to terminate the communication. This is used by the master to communicate with another slave or with the same slave in a different mode (transmit/receive mode) without releasing the bus.

8.2.5 STOP Signal

The master can terminate the communication by generating a STOP signal to free the bus. However, the master may generate a START signal followed by a calling command without generating a STOP signal first. This is called repeat START. A STOP signal is defined as a low to high transition of SDA while SCL is at a logical high; see Figure 8-2.

8.2.6 Arbitration Procedure

This interface circuit is a true multi-master system which allows more than one master to be connected. If two or more masters try to control the bus at the same time, a clock synchronization procedure determines the bus clock. The clock low period is equal to the longest clock low period among the masters; and the clock high period is the shortest among the masters. A data arbitration procedure determines the priority. A master will lose arbitration if it transmits a logic “1” while the others transmit logic “0”, the losing master will immediately switch over to slave receive mode and stops its data and clock outputs. The transition from master to slave mode will not generate a STOP condition. Meanwhile, a software bit will be set by hardware to indicate loss of arbitration.

8.2.7 Clock Synchronization

Since wire-AND logic is performed on the SCL line, a high to low transition on SCL line will affect the devices connected to the bus. The devices start counting their low period and once a device's clock has gone low, it will hold the SCL line low until the clock high state is reached. However, the change of low to high in this device clock may not change the state of the SCL line, if another device clock is still in its low period. Therefore synchronized clock SCL will be held low by the device which releases SCL to a logic high in the last place. Devices with shorter low periods enter a high wait state during this time (see Figure 8-3). When all devices concerned have counted off their low period, the synchronized clock SCL line will be released and go high. All of them will start counting their high periods. The first device to complete its high period will again pull the SCL line low.

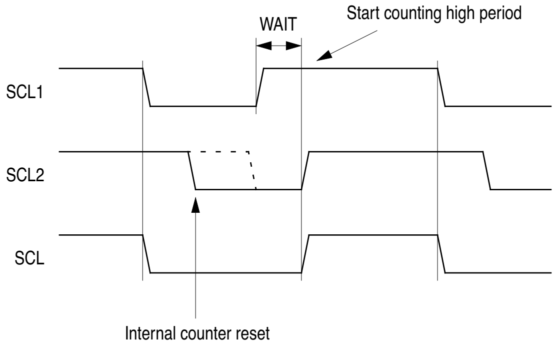


Figure 8-3 Clock Synchronization

8.2.8 Handshaking

The clock synchronization mechanism can be used as a handshake in data transfer. Slave device may hold the SCL low after completion of one byte transfer (9 bits). In such case, it will halt the bus clock and force the master clock in a wait state until the slave releases the SCL line.

8.3 M-Bus Registers

There are five registers used in the M-Bus interface, these are discussed in the following paragraphs.

8.3.1 M-Bus Address Register (MADR)

Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
$0039	MAD7	MAD6	MAD5	MAD4	MAD3	MAD2	MAD1		0000 0000

MAD1-MAD7 are the slave address bits of the M-Bus module.

8.3.2 M-Bus Frequency Register (MFDR)

Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
$003A				FD4	FD3	FD2	FD1	FD0	0000 0000

FD0-FD4 are used for clock rate selection. The serial bit clock frequency is equal to the CPU clock divided by the divider shown in Table 8-1.

Table 8-1 M-Bus Prescaler

FD4	FD3	FD2	FD1	FD0	DIVIDER
0	0	0	0	0	22
0	0	0	0	1	24
0	0	0	1	0	28
0	0	0	1	1	34
0	0	1	0	0	44
0	0	1	0	1	48
0	0	1	1	0	56
0	0	1	1	1	68
0	1	0	0	0	88
0	1	0	0	1	96
0	1	0	1	0	112
0	1	0	1	1	136
0	1	1	0	0	176
0	1	1	0	1	192
0	1	1	1	0	224
0	1	1	1	1	272
1	0	0	0	0	352
1	0	0	0	1	384
1	0	0	1	0	448
1	0	0	1	1	544
1	0	1	0	0	704
1	0	1	0	1	768
1	0	1	1	0	896
1	0	1	1	1	1088
1	1	0	0	0	1408
1	1	0	0	1	1536
1	1	0	1	0	1792
1	1	0	1	1	2176
1	1	1	0	0	2816
1	1	1	0	1	3072
1	1	1	1	0	3584
1	1	1	1	1	4352

For a 4MHz external crystal operation (2MHz internal operating frequency), the serial bit clock frequency of M-Bus ranges from 460Hz to 90,909Hz.

8.3.3 M-Bus Control Register (MCR)

Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
$003B	MEN	MIEN	MSTA	MTX	TXAK		SIFC	SIIC	0000 0000

Register bit definitions:

MEN - M-Bus Enable

1 (set) – M-Bus interface system enabled.

0 (clear) – M-Bus interface system disabled.

MIEN - M-Bus Interrupt Enable

1 (set) – M-Bus interrupt enabled.

0 (clear) – M-Bus interrupt disabled.

This bit enables the MIF (in MSR) for M-Bus interrupts.

MSTA - Master/Slave Select

1 (set) – M-Bus is set for master mode operation.

0 (clear) – M-Bus is set for slave mode operation.

Upon reset, this bit is cleared. When this bit is changed from 0 to 1, a START signal is generated on the bus, and the master mode is selected. When this bit is changed from 1 to 0, a STOP signal is generated and the operation mode changes from master to slave. In master mode, a bit clear immediately followed by a bit set of this bit generates a repeated START signal without generating a STOP signal.

MTX - Transmit/Receive Mode Select

1 (set) – M-Bus is set for transmit mode.

0 (clear) – M-Bus is set for receive mode.

TXAK - Acknowledge Enable

1 (set) – Do not send acknowledge signal.

0 (clear) – Send acknowledge signal at 9th clock bit.

If cleared, an acknowledge signal will be sent out to the bus at the 9th clock bit after receiving one byte of data. If set, no acknowledge signal response. This is an active low control bit.

8.3.4 M-Bus Status Register (MSR)

Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
$003C	MCF	MAAS	MBB	MAL	SIF	SRW	MIF	RXAK	1000 0001

The MIF and MAL bits are software clearable; while the other bits are read only.

MCF - Data Transfer Complete

1 (set) – A byte transfer has been completed.

0 (clear) – A byte is being transfer.

When MCF is set, the MIF (M-Bus interrupt) bit is also set. An M-Bus interrupt is generated if the MIEN bit is set.

MAAS - Addressed as Slave

1 (set) – Currently addressed as a slave.

0 (clear) – Not currently addressed.

This MAAS bit is set when its own specific address (M-Bus Address register) matches the calling address. When MAAS is set, the MIF (M-Bus interrupt) bit is also set. An interrupt is generated if the MIEN bit is set. Then CPU needs to check the SRW bit and set its MTX bit accordingly. Writing to the M-Bus Control register clears this bit.

MBB - Bus Busy

1 (set) – M-Bus busy.

0 (clear) – M-Bus idle.

This bit indicates the status of the bus. When a START signal is detected, MBB is set. When a STOP signal is detected, it is cleared.

MAL - Arbitration Lost

1 (set) – Lost arbitration in master mode.

0 (clear) – No arbitration lost.

This arbitration lost flag is set when the M-Bus master loses arbitration during a master transmission mode. When MAL is set, the MIF (M-Bus interrupt) bit is also set. This bit must be cleared by software.

SRW - Slave R/W Select

1 (set) – Read from slave, from calling master

0 (clear) – Write to slave from calling master.

When MAAS is set, the R/W command bit of the calling address sent from the master is latched into this SRW bit. By checking this bit, the CPU can then select slave transmit/receive mode by configuring MTX bit of the M-Bus Control register.

MIF - M-Bus Interrupt

1 (set) – An M-Bus interrupt has occurred.

0 (clear) – An M-Bus interrupt has not occurred.

When this bit is set, an interrupt is generated to the CPU if MIEN is set. This bit is set when one of the following events occurs:

1) Completion of one byte of data transfer. It is set at the falling edge of the 9th clock - MCF set.
2) A match of the calling address with its own specific address in slave mode - MAAS set.
3) A loss of bus arbitration - MAL set.

This bit must be cleared by software in the interrupt routine.

RXAK - Receive Acknowledge

1 (set) – No acknowledgment signal detected.

0 (clear) – Acknowledgment signal detected after 8 bits data transmitted.

If cleared, it indicates an acknowledge signal has been received after the completion of 8 bits data transmission on the bus. If set, no acknowledge signal has been detected at the 9th clock. This is an active low status flag.

8.3.5 M-Bus Data I/O Register (MDR)

Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
$003D	MD7	MD6	MD5	MD4	MD3	MD2	MD1	MD0	uuuu uuuu

In master transmit mode, data written into this register is sent to the bus automatically, with the most significant bit out first. In master receive mode, reading of this register initiates receiving of the next byte data. In slave mode, the same function applies after it has been addressed.

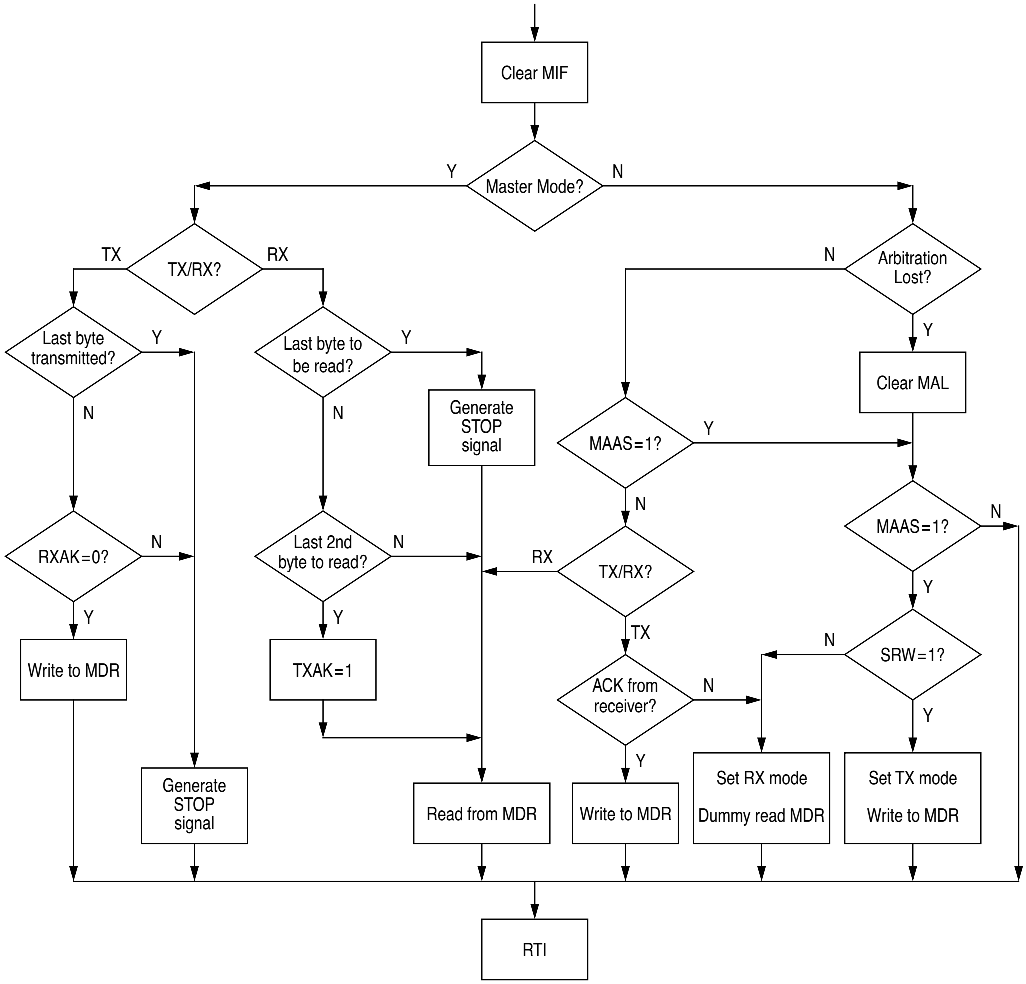


Figure 8-4 Flowchart of M-Bus Interrupt Routine

8.4 Programming Considerations

8.4.1 Initialization

Reset will put the M-Bus Control register to its default status. Before the interface can be used to transfer serial data, the following initialization procedure must be carried out.

1) Update Frequency Divider Register (MFDR) to select an SCL frequency.
2) Update M-Bus Address Register (MADR) to define its own slave address.
3) Set MEN bit of M-Bus Control Register (MCR) to enable the M-Bus interface system.
4) Modify the bits of M-Bus Control Register (MCR) to select Master/Slave mode, Transmit/Receive mode, interrupt enable or not.

8.4.2 Generation of a START Signal and the First Byte of Data Transfer

After completion of the initialization procedure, serial data can be transmitted by selecting the master transmit mode. If the device is connected to a multi-master bus system, the state of the M-Bus busy bit (MBB) must be tested to check if the serial bus is free. If the bus is free (MBB=0), the START condition and the first byte (the slave address) can be sent. An example program which generates the START signal and transmits the first data byte (slave address) is shown below:

```
             SEI                   ; DISABLE INTERRUPT
CHFLAG       BRSET  5,MSR,CHFLAG   ; CHECK THE MBB BIT OF THE
                                   ; STATUS REGISTER. IF IT IS
                                   ; SET, WAIT UNTIL IT IS CLEAR
TXSTART      BSET   4,MCR          ; SET TRANSMIT MODE
             BSET   5,MCR          ; SET MASTER MODE
                                   ; i.e. GENERATE START CONDITION
             LDA    #CALLING       ; GET THE CALLING ADDRESS
             STA    MDR            ; TRANSMIT THE CALLING
                                   ; ADDRESS
             CLI                   ; ENABLE INTERRUPT
```

8.4.3 Software Responses after Transmission or Reception of a Byte

Upon the completion of the transmission or reception of a data byte, the data transferring bit (MCF) will be set, indicating one byte communication has been finished. The M-Bus interrupt bit (MIF) will also be set to generate an M-Bus interrupt if the interrupt is enabled. Software must clear the

MIF bit in the interrupt routine first. The MCF bit can be cleared by reading the M-Bus Data I/O Register (MDR) in receive mode or writing to the MDR in transmit mode. Software may serve the M-Bus I/O in the main program by monitoring the MIF bit if the interrupt is disabled. The following is an example of a software response by a master in transmit mode in the interrupt routine (see Figure 8-4).

```
ISR             BCLR   1,MSR          ; CLEAR THE MIF FLAG
                BRCLR  5,MCR,SLAVE    ; CHECK THE MSTA FLAG,
                                      ; BRANCH IF SLAVE MODE
                BRCLR  4,MCR,RECEIVE ; CHECK THE MODE FLAG,
                                      ; BRANCH IF IN RECEIVE MODE
                BRSET  0,MSR,END      ; CHECK ACK FROM RECEIVER
                                      ; IF NO ACK, END OF
                                      ; TRANSMISSION
TRANSMIT        LDA    DATABUF        ; GET THE NEXT BYTE OF DATA
                STA    MDR            ; TRANSMIT THE DATA
```

8.4.4 Generation of the STOP Signal

A data transfer ends with a STOP signal generated by the master device. A master in transmit mode can simply generate a STOP signal after all the data have been transmitted. The following is an example showing how a STOP condition is generated by a master in transmit mode.

```
MASTX  BRSET  0,MSR,END     ; IF NO ACK, BRANCH TO END
       LDA    TXCNT         ; GET VALUE FROM THE
                            ; TRANSMITTING COUNTER
       BEQ    END           ; IF NO MORE DATA, BRANCH TO
                            ; END
       LDA    DATABUF       ; GET NEXT BYTE OF DATA
       STA    MDR           ; TRANSMIT THE DATA
       DEC    TXCNT         ; DECREASE THE TXCNT
       BRA    EMASTX        ; EXIT
END    BCLR   5,MCR         ; GENERATE A STOP CONDITION
EMASTX RTI                  ; RETURN FROM INTERRUPT
```

If a master receiver wants to terminate a data transfer, it must inform the slave transmitter by not acknowledging the last byte of data. This can be achieved by setting the transmit acknowledge bit (TXAK) before reading the 2nd last byte of data. Before reading the last byte of data, a STOP signal must be generated first. The following is an example showing how a STOP signal is generated by a master in receive mode.

```
MASR   DEC    RXCNT
       BEQ    ENMASR        ; LAST BYTE TO BE READ
       LDA    RXCNT
       DECA                 ; CHECK LAST 2ND BYTE TO BE READ
       BNE    NXMAR         ; NOT LAST ONE OR LAST SECOND
```

```
LAMAR  BSET   3,MCR         ; LAST SECOND, DISABLE ACK
                            ; TRANSMITTING
       BRA    NXMAR
ENMASR BCLR   5,MCR         ; LAST ONE, GENERATE 'STOP'
                            ; SIGNAL
NXMAR  LDA    MDR           ; READ DATA AND STORE
       STA    RXBUF
       RTI
```

8.4.5 Generation of a Repeated START Signal

At the end of data transfer, if the master still wants to communicate on the bus, it can generate another START signal followed by another slave address without first generating a STOP signal. A program example is as shown.

```
RESTART      BCLR   5,MCR         ; ANOTHER START (RESTART) IS
             BSET   5,MCR         ; GENERATED BY THESE TWO
                                  ; CONSECUTIVE INSTRUCTIONS
             LDA    #CALLING      ; GET THE CALLING ADDRESS
             STA    MDR           ; TRANSMIT THE CALLING
                                  ; ADDRESS
```

8.4.6 Slave Mode

In the slave service routine, the master addressed as slave bit (MAAS) should be tested to check if a calling of its own address has been received (Figure 8-4). If MAAS is set, software should set the transmit/receive mode select bit (MTX bit of MCR) according to the R/$\overline{\text{W}}$ command bit (SRW). Writing to the MCR clears the MAAS automatically. A data transfer may then be initiated by writing to MDR or a dummy read from MDR.

In the slave transmit routine, the received acknowledge bit (RXAK) must be tested before transmitting the next byte of data. RXAK, if set indicates the end of data signal from the master receiver, the slave transmitter must then switch from transmit mode to receive mode by software and a dummy read must follow to release the SCL line so that the master can generate a STOP signal.

8.4.7 Arbitration Lost

If more than one master want to acquire the bus simultaneously, only one master can win and the others will lose arbitration. The losing device immediately switches to slave receive mode by M-Bus hardware. Its data output to the SDA line is stopped, but internal transmit clock still runs until the end of the data byte transmission. An interrupt occurs when this dummy byte transmission is accomplished with MAL=1 and MSTA=0. If one master attempts to start transmission while the bus is being controlled by another master, the transmission will be inhibited; the MSTA bit will be changed from 1 to 0 without generating STOP condition; an interrupt will be generated and the MAL bit set to indicate that the attempt to acquire the bus has failed. Considering these cases, the slave service routine should test the MAL bit first, and software should clear the MAL bit if it is set.

8.5 Software Supported M-Bus Interface

Port pins PC6 and PC7 are designed with a hardware interrupt circuit for the software supported M-Bus interface to detect the “Start” condition of the M-Bus protocol. This interrupt uses the same interrupt vector as for the hardware M-Bus interrupts, at address $3FF2 and $3FF3. The user is responsible for determining the source of the M-Bus interrupt in the interrupt service routine by reading the flags.

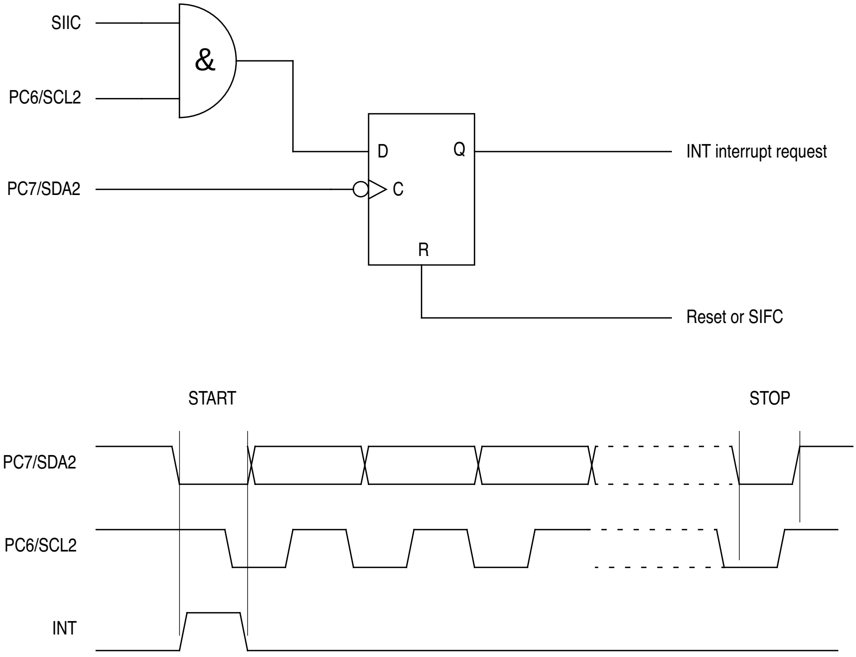


Figure 8-5 Software Supported M-Bus Interrupt

The software supported M-Bus interrupt related control and status bits are at the following registers:

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
MCR	$003B	MEN	MIEN	MSTA	MTX	TXAK		SIFC	SIIC	0000 0000

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
MSR	$003C	MCF	MAAS	MBB	MAL	SIF	SRW	MIF	RXAK	1000 0001

SIFC - Software M-Bus Interrupt Flag Clear

1 (set) – Clear software supported M-Bus interrupt flag in MSR.

0 (clear) – No effect.

SIIC - Software M-Bus Enable

1 (set) – If PC6 and PC7 satisfies the “Start” condition of the M-Bus protocol, SIF in the MSR will be set and M-Bus interrupt will be generated.

0 (clear) – PC6 and PC7 pins are configured for standard I/O operation.

SIF - Software Supported M-Bus Interrupt Flag

1 (set) – Software M-Bus interrupt has occurred.

0 (clear) – No software M-Bus interrupt has occurred.

THIS PAGE LEFT BLANK INTENTIONALLY

9
SYNC SIGNAL PROCESSOR

This section describes the operation of the SSP module in the MC68HC05BS8.

9.1 Introduction

The functions of the SSP include the following:

- Polarity correction
- Sync separation
- Sync pulse reshaping
- Sync pulse detection
- Horizontal line counting
- Vertical frequency counting
- Free running signals generation

In addition, an interrupt can be generated for each vertical frame at a user specified horizontal line number. The SSP accepts composite signals from a video processor or separate sync inputs from a host computer.

For separate sync inputs, the HTTL output is identical to the incoming horizontal sync with negative sync polarity. The VTTL output is triggered by the leading edge of the incoming sync pulse, and the sync pulse reshaper will correct the pulse width to a certain time period regardless of the incoming sync width. Reassembled sync pulses are inserted for HTTL signal during the vertical sync period for composite sync input.

The HSYNC, VSYNC, and CSYNC inputs have internal filters to improve noise immunity. Any pulse that is shorter than 1.5 internal bus clock periods will be regarded as a glitch, and will be ignored.

Note: All quoted timings in this section are based on an internal bus frequency of 2MHz, i.e. t_{CYC}=0.5µs, unless otherwise stated.

9.2 Polarity Correction

The polarity correction block of the sync signal processor accepts the input sync signals and converts them to negative polarity signals, regardless of the polarity of the inputs. The following describes the methodologies used in polarity correction.

9.2.1 Separate Vertical Sync Input

To test the polarity of the input sync signal, the duration of the low pulse is examined. If the low period is longer than a specific value (512µs or $1024t_{CYC}$), as in the case of positive polarity input sync, the input sync will be inverted before output. For negative polarity input sync signal, it is anticipated that the duration of the low pulse would be shorter than the specific value, and the input sync signal passes through to the output without inversion.

This polarity correction is a continuous process, and the error margin is equal to the maximum permissible sync pulse width specified (512µs or $1024t_{CYC}$). At power-up or system reset, negative polarity at input is assumed.

9.2.2 Separate Horizontal or Composite Sync Input

Since the input at HSYNC can be either a pure horizontal sync signal or a composite sync signal, different methodologies are used in polarity correction.

Unlike the polarity correction for VSYNC, both the high pulse and low pulse of the sync signal at HSYNC are examined. If the pulse, either active high or low, is longer than a certain period (8µs or 16 t_{CYC}), it will be regarded as a long pulse. If there are 8 consecutive low long pulses, the input sync signal will be confirmed as a positive polarity sync signal, and will be inverted. If there are 8 consecutive high long pulses, it will be confirmed as a negative polarity sync signal.

The operation of this module is also continuous, and the error margin is equal to the period of the pre-set number (default is 8) of horizontal sync pulses. At power-up or system reset, negative polarity at input is assumed.

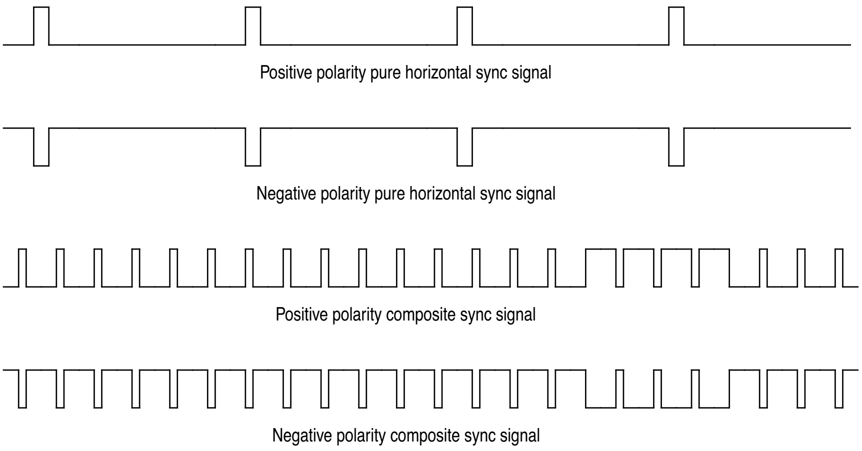

Positive polarity pure horizontal sync signal
Negative polarity pure horizontal sync signal
Positive polarity composite sync signal
Negative polarity composite sync signal

Figure 9-1 Sync Signal Polarity Correction

9.3 Sync Detection

The sync detector determines whether the incoming sync signal is active. Both sync high and low pulse widths must be within the specific values to be regarded as active. The respective HDET and/or VDET flags will be set if the HSYNC and VSYNC signals are active.

9.4 Free-running Pseudo Sync Signal Generator

If no active sync signals are detected, a free-running sync signal generator will be enabled. It generates a pseudo vertical sync at 63.78Hz (1/(t_{CYC} x 31360)) and a pseudo horizontal sync at 57.14KHz (1/(t_{CYC} x 35)). This set of free running sync signals replaces the inactive sync signals at the inputs and will be fed to the VTTL and HTTL pins if the pins are selected for VTTL and HTTL function.

If both vertical and horizontal sync signals are detected (VDET and HDET are set), the SOUT bit must be set in order to output the processed signals.

9.5 Sync Separation

Figure 9-2 is a block diagram of the Sync Separator which includes the duration counters for the high and low pulses, a counter for the number of valid horizontal sync pulses, a register to hold the number of horizontal lines per frame, a logic block for horizontal and vertical sync pulse separation, a comparator, and a sync pulse insertion circuit.

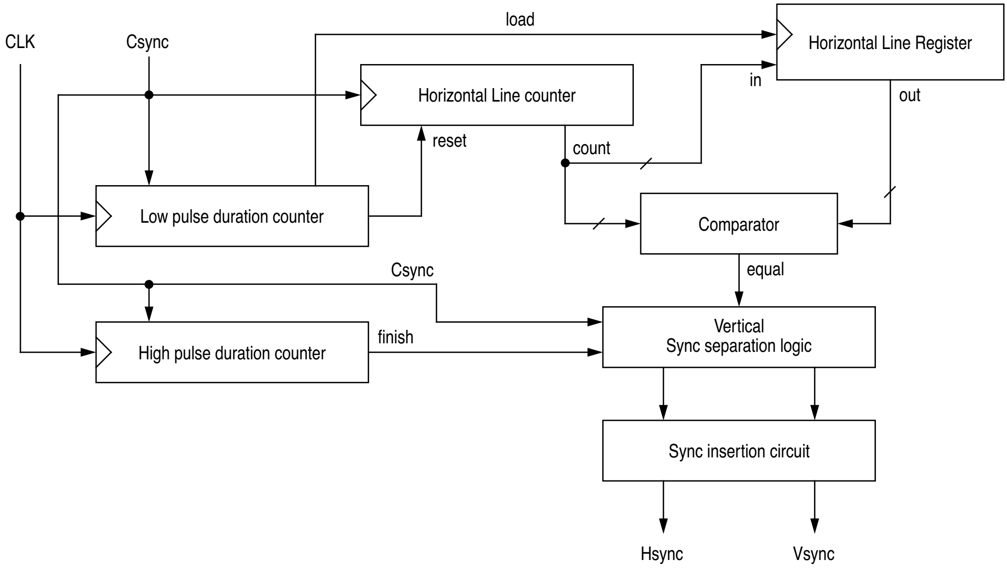


Figure 9-2 Sync Separator

The Low pulse duration counter examines the low pulse width of the incoming composite sync signal. If it is within the horizontal sync pulse limit (8µs or 16 t_{CYC}), a horizontal sync pulse is detected, and the horizontal line counter is advanced. If the low pulse is wider than the limit, a vertical sync pulse is detected, and the content of the Horizontal line counter is loaded into the Horizontal Line register. The Low pulse duration counter then resets the Horizontal line counter.

The High Pulse Duration Counter examines the high pulse width of the incoming composite sync signal. If it is longer than a specific value (8µs or 16 t_{CYC}), the vertical sync pulse has finished and a “finish” signal will be given to the Sync Separation Logic.

Sync Separation Logic passes the Csync signal to the Hsync output until there is an “equal” signal from the comparator. The Hsync output will then output an reassembled waveform by the Sync Insertion Circuit to emulate the Hsync pulses, and the Vsync output is set to “low” at the coming falling edge of the Csync signal. After the “finish” signal has been sensed, the Vsync output is fixed to “high”, and the Hsync output follows the Csync input again.

9.6 Vertical Sync Pulse Reshaper

The vertical sync pulse width at VTTL output is formatted by the Vertical Sync Pulse Reshaper, such that, it's falling edge follows the input signal's falling edge, and it's rising edge is at the 6th falling edge of the input horizontal sync signal (HSYNC). Notice that, if the input signal is a composite signal, the VTTL pulse width will be longer, for there may not be any falling edge during the vertical sync pulse period. Figure 9-3 shows the different in VTTL pulse widths for different input signal formats.

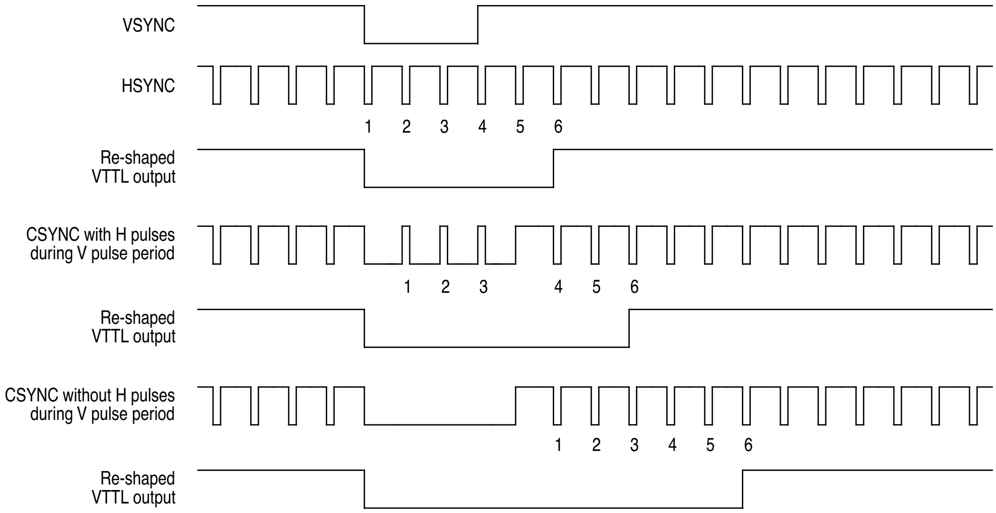
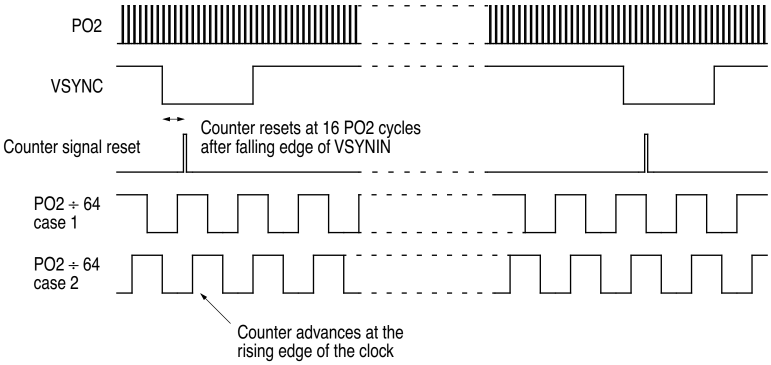


Figure 9-3 VTTL Pulse Widths For Different Input Signal Formats

9.7 Sync Signal Counters

There are two counters (horizontal line counter and vertical frequency counter) to count the number of horizontal sync pulses and the number of system clock cycles between two vertical sync pulses. These two data can be read by the CPU to check the signal frequencies and can be used to determine the video mode. Notice that the value in the vertical frequency counter will be subtracted by 240 before loading into the Vertical Frequency register. In this way, the 9-bit register can cover a vertical frequency ranged from 42Hz to 130Hz. Figure 9-4 shows a more detailed block diagram of these counters. Figure 9-4 shows the vertical frequency counter timings. It indicates that there will be ±1 count error on the reading from the register for the same vertical frequency.


PO2
VSYNC
Counter signal reset
Counter resets at 16 PO2 cycles after falling edge of VSYNIN
PO2 ÷ 64 case 1
PO2 ÷ 64 case 2
Counter advances at the rising edge of the clock

1. The value of the counter will be loaded into the register before it is reset.
2. The Vertical Frequency Counter is clocked by a PO2 ÷ 64 clock.
3. Because of the asynchronous nature between PO2 and VSYNIN, the register will have one more count in case 2 than in case 1.

Figure 9-4 Vertical Frequency Counter Timing

9.8 VSYNC Interrupt

The CPU will process an Sync Signal Processor VSYNC interrupt if the following conditions are satisfied:

1) the I-bit of the CCR is cleared,

2) the VSIE bit of the Interrupt Line Count register (ILCR) is set, and

3) the value of the horizontal line counter matches the value set in the ILCR.

This interrupt will vector to the interrupt service routine located at the address specified by the contents of $3FF8 and $3FF9. The VSYNC interrupt latch will be cleared automatically by fetching of these vectors.

This allows an interrupt to be generated in each vertical frame after a certain number of lines (0-127) to check the status of the monitor and conditions.

9.9 Sampling Pulse Output

The circuit is responsible for generating the SAM signal for the Video Chip set. The SAM signal is a sampling signal, which outputs a positive pulse at the Vsync pulse, and outputs another positive pulse when the value of the Sampling Pulse register matches the horizontal line counter. The pulse width is equal to one horizontal line period.

9.10 SSP Registers

There are six registers associated with the Sync Signal Processor, these are described below.

9.10.1 Sync Signal Control and Status Register (SSCSR)

Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
$000A	VPOL	HPOL	VDET	HDET	SOUT	INSRT	SIN1	SIN0	0000 0000

VPOL - Vertical Sync Input Polarity

1 (set) – VSYNC input is positive polarity.

0 (clear) – VSYNC input is negative polarity.

Vertical Sync Input Polarity flag indicates the polarity of the incoming signal at the VSYNC input.

HPOL - Horizontal Sync Input Polarity

1 (set) – HSYNC input is positive polarity.

0 (clear) – HSYNC input is negative polarity.

Horizontal Sync Input Polarity flag indicates the polarity of the incoming signal at the HSYNC input.

VDET - Vertical Sync Signal Detect

1 (set) – An active vertical sync is detected at VSYNC input.

0 (clear) – No vertical sync signal at VSYNC input; use internal generated Vsync for VTTL.

This bit is set when an active vertical sync signal is detected on the VSYNC pin. If cleared, it indicates there is no active signal, and the VTTL will output the internally generated Vsync signal. An active vertical sync signal is defined as:

$$\text{VDET} = (\text{VSYNC pulse width} < 1024t_{CYC}) \cdot (15.36\text{x}10^3 t_{CYC} < \text{VSYNC period} < 48.128\text{x}10^3 t_{CYC})$$

HDET - Horizontal Sync Signal Detect

1 (set) – An active horizontal sync is detected at HSYNC input.

0 (clear) – No horizontal sync signal at HSYNC input; use internal generated Hsync for HTTL.

This bit is set when an active horizontal sync signal is detected on the HSYNC pin. If cleared, it indicates there is no active signal, and the HTTL will output the internally generated Hsync signal. An active horizontal sync signal is defined as:

$$HDET=(\text{HSYNC pulse width} < 16t_{CYC}) \cdot (\text{HSYNC period} < 128t_{CYC}) \cdot [(\text{H line per frame} < 4096) + (VDET=0)]$$

SOUT - Sync Output Select

1 (set) – Use processed VSYNC and HSYNC inputs for VTTL and HTTL.

0 (clear) – Use internally generated sync signals for VTTL and HTTL.

When cleared, the outputs to VTTL and HTTL are the internally generated signals. When set, the outputs are the processed input signals. This bit can only be set if both VDET and HDET are logic 1’s, and will be cleared automatically if VDET or HDET is not logic “1”. Reset clears this bit.

INSRT - Hsync Insertion

1 (set) – No inserted pulses. HTTL will always follow the HSYNC input.

0 (clear) – For composite sync inputs, emulated sync pulses will be inserted into the HTTL signal during the vertical sync pulse.

For separate sync inputs, when this Hsync Insertion bit is cleared, sync pulses will continue to be the Hsync signal during the vertical sync pulse. For composite sync input, when this bit is cleared, emulated sync pulses will be inserted into the HTTL during the vertical sync pulse. In both cases, when this bit is set, there will be no inserted pulses, and HTTL will always follow the HSYNC input. Reset clears this bit.

SIN1:SIN0 - Sync Input Source

These two bits selects the source of the input sync signals. Reset clears these bits.

SIN1	SIN0	Sync input source
0	0	Separated sync signal through VSYNC and HSYNC inputs.
0	1	Composite sync signal through HSYNC input.
1	X	Composite sync signal through CSYNC input.

9.10.2 Vertical Frequency Register (VFR)

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
VFR	$000B	VF7	VF6	VF5	VF4	VF3	VF2	VF1	VF0	0000 0000

This 9-bit (the 9th bit is in bit 7 of LFHR at $0C) read only register is used to calculate the vertical frame frequency. A 10-bit counter counts the number of internal clocks between two VSYNC pulses. The counted value will then be transferred to this register. The data corresponds to the period of one vertical frame. This register can be read to determine if the frame frequency is valid, and to determine the video mode. Note that data is valid only if VDET = 1.

The frame frequency is calculated by $1/((\mathrm{VFR}\pm 1 + 240)\,\mathrm{x}\,16t_{CYC})$.

Table 9-1 shows sample values for the Vertical Frequency register, all VFR numbers are in hexadecimal**.**

Table 9-1 Vertical Frame Frequencies

VFR	Min. Freq.	Max. Freq.
$004	127.6	128.6
$005	127.0	128.1
$006	126.5	127.6
$007	126.0	127.0
$1FA	41.8	41.9
$1FB	41.8	41.8
$1FC	41.7	41.8
$1FD	41.7	41.8

9.10.3 Line Frequency Registers (LFRs)

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
LFHR	$000C	VF8				LF11	LF10	LF9	LF8	0000 0000
LFLR	$000D	LF7	LF6	LF5	LF4	LF3	LF2	LF1	LF0	0000 0000

This 12-bit read only register pair contains the number of lines in each vertical frame. An internal line counter counts the number of horizontal sync pulses between two vertical sync pulses and then transfers the counted value to this register. The data can be read to determine if the line frequency is valid, and to determine the video mode. Note that data is valid only if HDET = VDET = 1.

The VF8 bit is the ninth bit of the Vertical Frequency register (VFR).

9.10.4 Interrupt Line Count Register (ILCR)

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
ILCR	$000E	VSIE	LC6	LC5	LC4	LC3	LC2	LC1	LC0	0000 0010

This is a read/write register containing the line number for which the Vertical Sync Interrupt is to be generated. Interrupt will be generated if VSIE bit is set, I bit in CCR is cleared, and the internal line counter value matches the setting in this register after the Vsync pulse. The Vsync Interrupt Vectors are at $3FF8 and $3FF9, and the interrupt latch is cleared by fetching the interrupt vectors.

VSIE - Vsync Interrupt Enable

This bit enables and disables the Vsync interrupt.

1 (set)	–	Vsync interrupt enabled.
0 (clear)	–	Vsync interrupt disabled.

LC6:0 - Line Count for Vsync Interrupt

These 7 bits store the line number for which the Vsync Interrupt will occur. The number is ranged from 0 to 127.

9.10.5 Sampling Pulse Register (SPR)

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
SPR	$000F		SP6	SP5	SP4	SP3	SP2	SP1	SP0	0000 0010

This read/write register contains the line number for which the sampling pulse in SAM output to be generated. The line number is ranged from 0 to 127. Sampling pulses are produced when there is a Vsync pulse, or this register matches the horizontal line counter.

9.11 System Operation

The sync processor accepts sync signals from the main computer; the signals can either be separate Hsync and Vsync or composite sync through HSYNC input. Polarity correction is performed before the sync signals go any further into the system. The sync pulse detection blocks will continuously monitor the signal, to see if it is active. If the signal is not active, the circuit will switch to output the internally generated clock signal. This will protect the circuits behind from being damaged by an inactive signal.

Figure 9-5 shows an example of SSP operation.

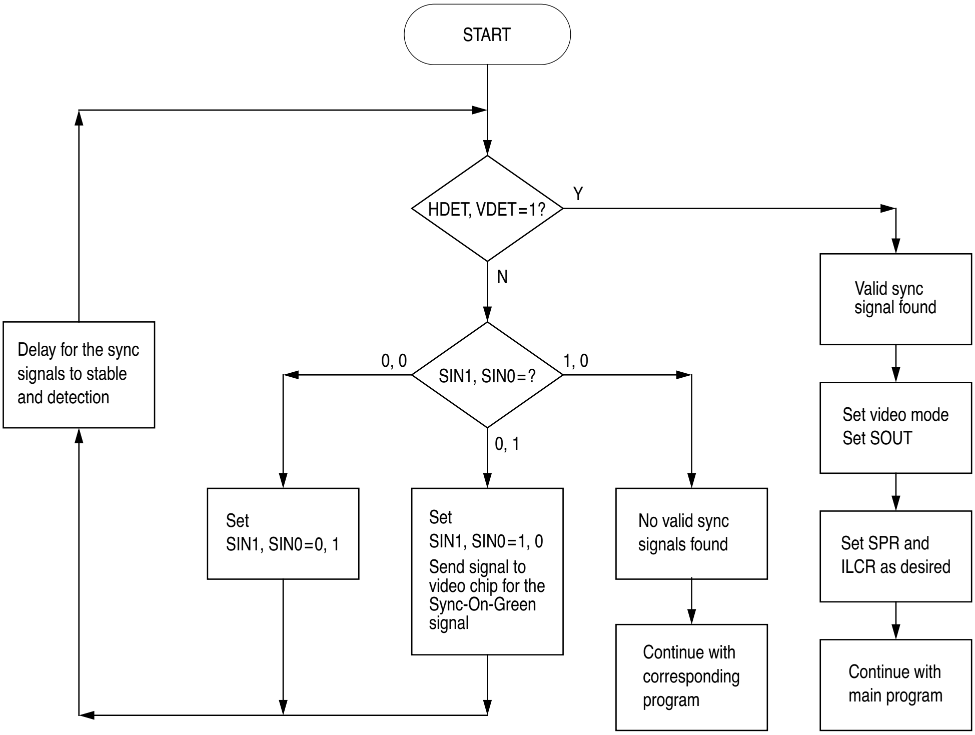

START
HDET, VDET=1?
Y
N
Valid sync signal found
Delay for the sync signals to stable and detection
0, 0
SIN1, SIN0=?
1, 0
0, 1
Set video mode
Set SOUT
Set SIN1, SIN0=0, 1
Set SIN1, SIN0=1, 0
Send signal to video chip for the Sync-On-Green signal
No valid sync signals found
Set SPR and ILCR as desired
Continue with corresponding program
Continue with main program

Figure 9-5 Example of SSP operation

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 MC68HC05BS8.

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.

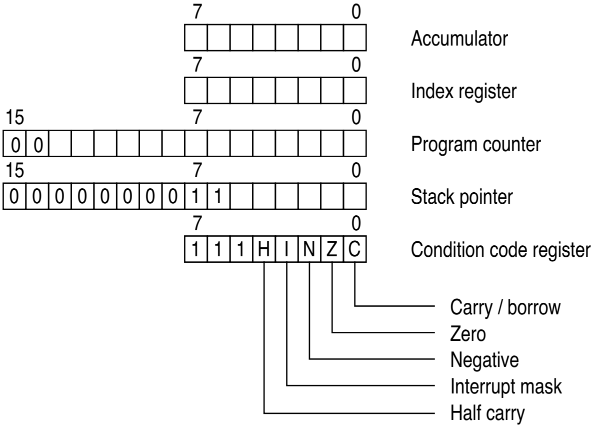


Figure 10-1 Programming model

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.

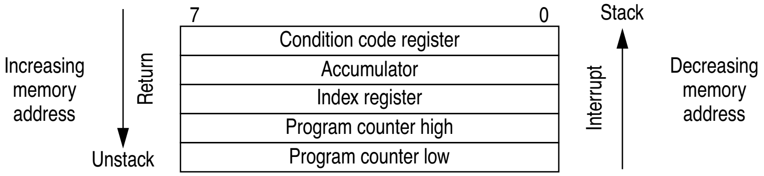


Figure 10-2 Stacking order

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

10

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

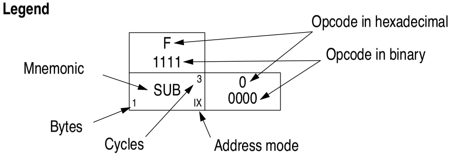
	Bit manipulation		Branch	Read/modify/write					Control		Register/memory						
	BTB	BSC	REL	DIR	INH	INH	IX1	IX	INH	INH	IMM	DIR	EXT	IX2	IX1	IX	
High / Low	0 0000	1 0001	2 0010	3 0011	4 0100	5 0101	6 0110	7 0111	8 1000	9 1001	A 1010	B 1011	C 1100	D 1101	E 1110	F 1111	High / Low
0 0000	BRSET0 5 3 BTB	BSET0 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 MC68HC05BS8 has two low-power operating modes: the STOP mode and WAIT mode. These two modes help to reduce the power required for the MCU by stopping various internal clocks and/or the on-chip oscillator. The flow of the STOP and WAIT modes is shown in Figure 11-1.

11.1 STOP Mode

The STOP instruction places the MCU in its lowest power consumption mode. In the STOP mode the internal oscillator is turned off, halting all internal processing.

When the CPU enters STOP mode, the I-bit in the Condition Code Register will be cleared automatically. This enables the external hardware interrupt to “wake-up” the MCU. All other registers and memory remain unchanged. All input/output lines remain unchanged.

The MCU can be brought out of the STOP mode only by a hardware interrupt or an externally generated reset (POR or $\overline{\text{RESET}}$). When exiting the STOP mode the internal oscillator will resume after a 4064 internal processor clock cycle oscillator stabilization delay.

Note: The STOP mode cannot be entered if the COP watchdog timer is running.

11.2 WAIT Mode

The WAIT instruction places the MCU in a low power mode. In the WAIT mode the internal processor clock is halted, suspending all processor and internal bus activity. Other internal clocks remain active, permitting interrupts to be generated from the sub-systems, or a reset generated from the COP Watchdog Timer. The Timer may be used to generate a periodic exit from the WAIT mode. Execution of the WAIT instruction automatically clears the I-bit in the Condition Code Register, so that any hardware interrupt can “wake-up” the MCU. All other registers, memory, and input/output lines remain in their previous states.

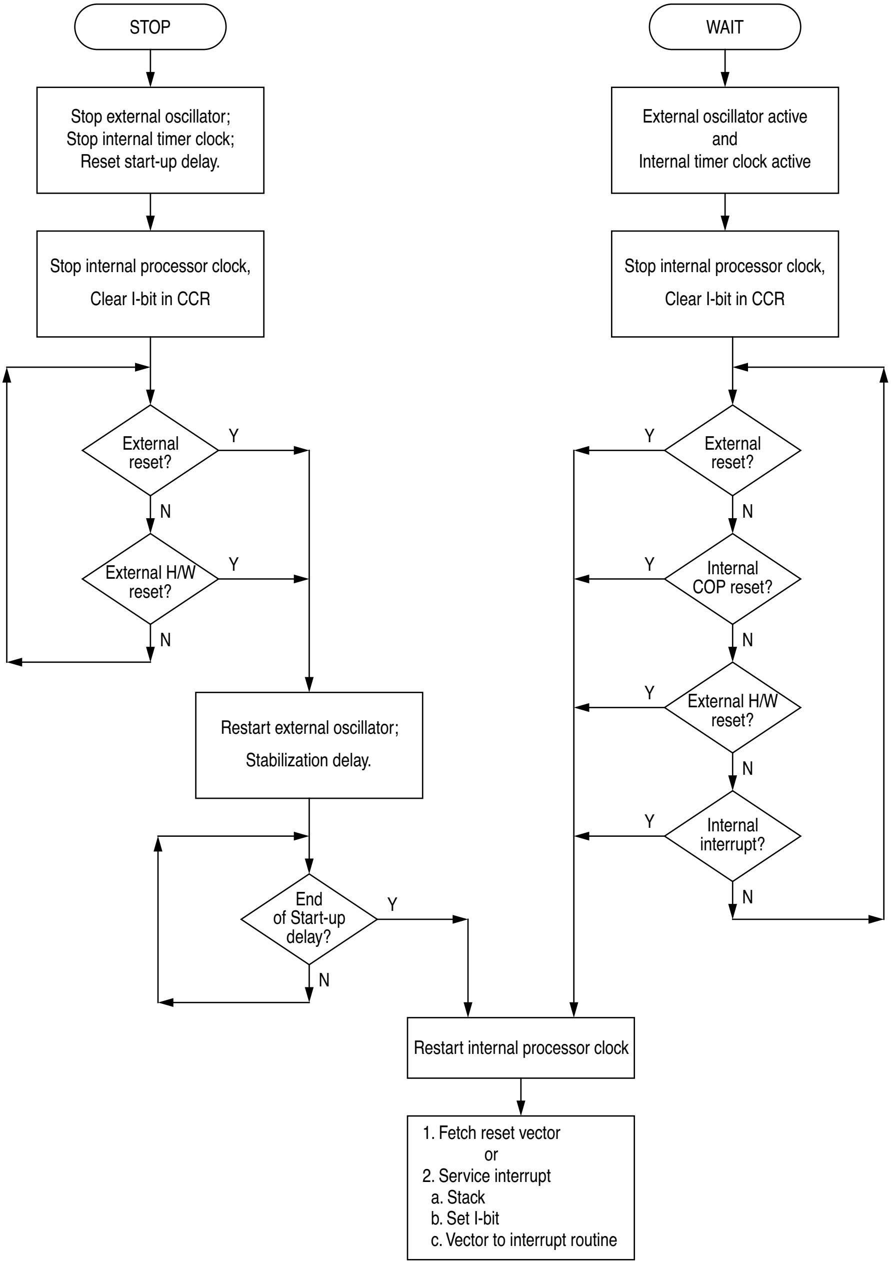


Figure 11-1 STOP and WAIT Flowchart

11.3 Data Retention Mode

The contents of RAM and CPU registers are retained at supply voltages as low as 2.0V dc. This is called the data-retention mode where the data is held, but the device is not guaranteed to operate. The $\overline{\text{RESET}}$ pin must be held low during data-retention mode.

11.4 COP Watchdog Timer Considerations

The COP Watchdog Timer is enabled by default after a reset, and can be disabled by writing a “0” to the COP bit in the Option register (bit 6 at address $1D). Once enabled, any execution of the STOP instruction will be executed as a WAIT instruction. That is, the STOP mode cannot be entered if the COP Watchdog Timer is enabled.

If the COP Watchdog Timer is enabled, the COP will reset the MCU when it times out. For a system that must have intentional uses of the WAIT mode, care must be taken to prevent such situations from happening during normal operations by arranging timely interrupts to reset the COP Watchdog Timer.

THIS PAGE LEFT BLANK INTENTIONALLY

12
OPERATING MODES

The MC68HC05BS8/*MC68HC705BS8* 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.

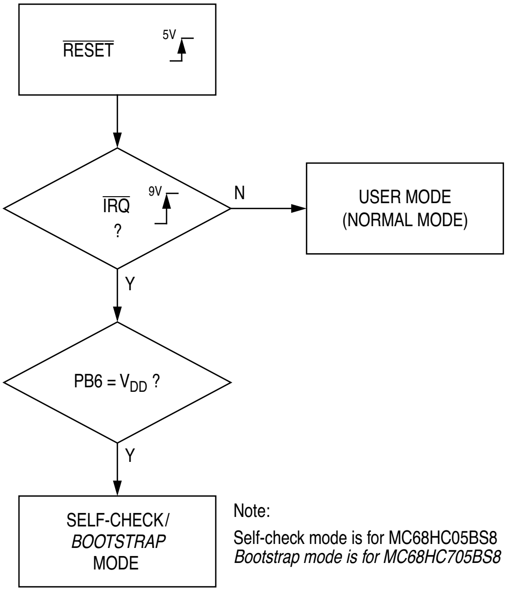


Figure 12-1 Flowchart of Mode Entering

Table 12-1 Mode Selection

$\overline{\text{RESET}}$	$\overline{\text{IRQ}}$	PB6	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 operating mode of the MC68HC05BS8/*MC68HC705BS8* 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 self-check mode is provided on the MC68HC05BS8 for the user to check device functions with an on-chip self-check program masked at location $1600 to $17FF under minimum hardware support. The self-check schematic is shown in Figure 12-3. Figure 12-2 is the criteria to enter self-check mode, where PB6's condition is latched within first two clock cycles after the rising edge of the reset. PB6 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.

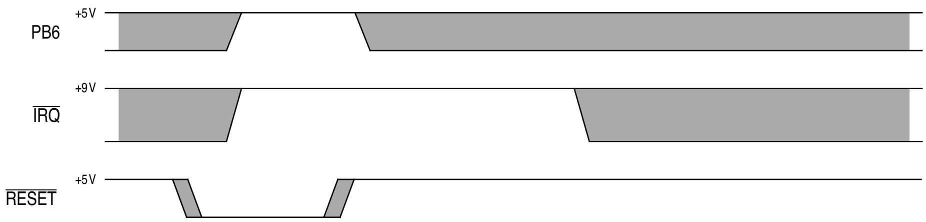


Figure 12-2 Self-Check Mode Timing

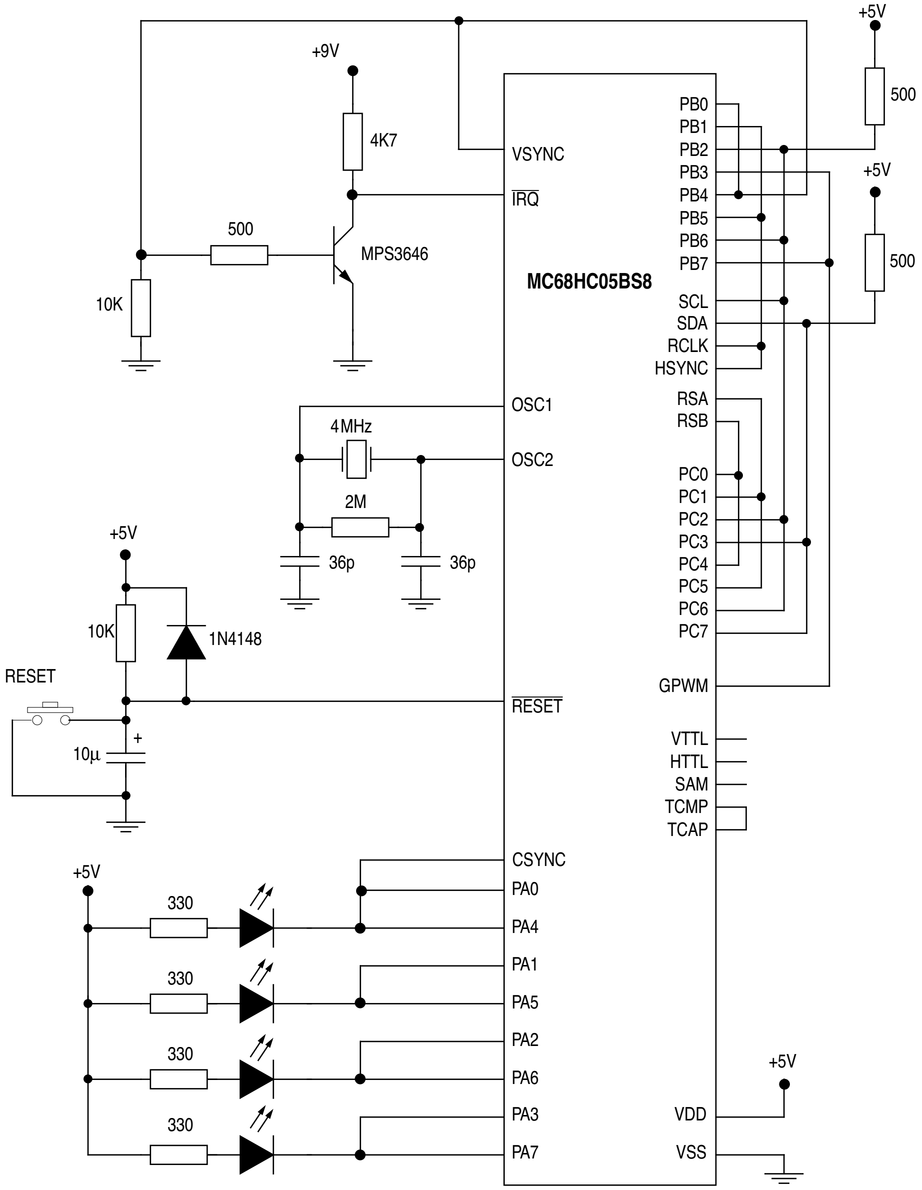


Figure 12-3 MC68HC05BS8 Self-Test Circuit

Table 12-2 Self-Check Report

PA4	PA3	PA1	PA0	REMARKS
Flashing				O.K. (self-check is on-going)
1	1	1	1	Bad port A
1	1	1	0	Bad port B
1	1	0	1	Bad port C
1	1	0	0	Bad RAM
1	0	1	1	Bad ROM
1	0	1	0	Bad $\overline{\text{IRQ}}$
1	0	0	1	Bad KBI
0	1	1	1	Bad CTimer
0	1	1	0	Bad GPWM

1=LED off, 0=LED on

12.3 Bootstrap Mode

The bootstrap mode is provided in the EPROM part (MC68HC705BS8) 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 PB6 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. The user EPROM consists of 10K-bytes, from location $1800 to $3FDF.

Refer to Appendix A for further details on MC68HC705BS8.

13
ELECTRICAL SPECIFICATIONS

This section contains the electrical specifications for MC68HC05BS8.

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
$\overline{\text{IRQ}}$	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 V_{IN} and V_{OUT} be constrained to the range $V_{SS} \leq (V_{IN}$ or $V_{OUT}) \leq V_{DD}$. Reliability of operation is enhanced if unused inputs are connected to an appropriate logic voltage level (e.g. either V_{SS} or V_{DD}).

13.2 Thermal Characteristics

CHARACTERISTICS	SYMBOL	VALUE	UNIT
Thermal resistance - Plastic 44-pin QFP package	θ_{JA}	60	°C/W

13

13.3 DC Electrical Characteristics

Table 13-1 DC Electrical Characteristics for MC68HC05BS8

(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 I_{LOAD} = +10µA	 V_{OH} V_{OL}	 V_{DD}–0.1 —	 — —	 — 0.1	 V V
Output high voltage (I_{LOAD}=–8mA) PA0-PA7, PB0-PB1, PC0-PC7, TCMP (TTL level) VTTL, HTTL, SAM	V_{OH}	 V_{DD}–0.8 2.4	 — —	 — —	V V
Output low voltage (I_{LOAD}=+16mA) PA0-PA7, PC0-PC7, VTTL, HTTL, SAM, TCMP, SCL, SDL (I_{LOAD}=+10mA) PB0-PB7	V_{OL}	—	—	0.4	V
Input high voltage PA0-PA7, PB0-PB7, PC0-PC7, TCAP, SCL, SDA, $\overline{\text{IRQ}}$, $\overline{\text{RESET}}$, OSC1 (TTL level) VSYNC, HSYNC, CSYNC	V_{IH}	0.7xV_{DD} 2.0	— —	V_{DD} V_{DD}	V V
Input low voltage PA0-PA7, PB0-PB5, PC0-PC7, TCAP, SCL, SDA, $\overline{\text{IRQ}}$, $\overline{\text{RESET}}$, OSC1 (TTL level) VSYNC, HSYNC, CSYNC	V_{IL}	V_{SS} V_{SS}	— —	0.2xV_{DD} 0.8	V V
Supply current: Run Wait Stop 25°C 0°C to 70°C (Standard)	I_{DD}	 — — — —	 2.5 1.3 1.6 5.0	 5.8 2.0 30 50	 mA mA µA µA
I/O ports high-Z leakage current PA0-PA7, PB0-PB5, PC0-PC7, SDA, SCL, GPWM, RSA, RSB	I_{IL}	—	—	±10	µA
Input current $\overline{\text{IRQ}}$, $\overline{\text{RESET}}$, OSC1, VSYNC, HSYNC, TCAP	I_{IN}	—	—	1	µA
Capacitance ports (as input or output), $\overline{\text{RESET}}$, $\overline{\text{IRQ}}$, OSC1, OSC2, TCAP SDA, SCL, HSYNC, VSYNC, CSYNC	 C_{OUT} C_{IN}	 — —	 — —	 12 8	 pF pF
Low voltage reset threshold voltage	V_{LVR}	2.6	2.8	3.0	V

Notes:

(1) All values shown reflect average measurements.

(2) Typical values at midpoint of voltage range, 25°C only.

(3) Wait I_{DD}: only timer system active.

(4) Run (operating) I_{DD}, Wait I_{DD}: measured using external square wave clock source to OSC1 (f_{OSC}=4.0MHz), all inputs 0.2 Vdc from rail; no dc loads, less than 50pF on all outputs, C_L=20pF on OSC2.

(5) Wait, Stop I_{DD}: all ports configured as inputs, V_{IL}=0.2 Vdc, V_{IH}=V_{DD} – 0.2 Vdc.

(6) Stop I_{DD}: measured with OSC1 = V_{SS}.

(7) Wait I_{DD} is affected linearly by the OSC2 capacitance.

(8) $\overline{\text{IRQ}}$ should be pulled high at all times.

13.4 Control Timing

Table 13-2 Control Timing

(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	4.4 4.4	MHz MHz
Internal operating frequency (f_{OSC}/2) Crystal External clock	f_{OP}	– dc	2.2 2.2	MHz MHz
Processor cycle time	t_{CYC}	450	–	ns
Crystal oscillator start-up time (crystal option)	t_{OXON}	–	100	ms
Stop Recovery Start-up Time (crystal option)	t_{ILCH}	–	100	ms
External RESET pulse width	t_{RL}	1.5	–	t_{CYC}
Power-on RESET output width (4064 cycles)	t_{PORL}	4064	–	t_{CYC}
Watchdog RESET output pulse width	t_{DOGL}	1.5	–	t_{CYC}
Watchdog time-out	t_{DOG}	2^{14} x 7	2^{17} x 7	t_{CYC}
Timer resolution (see note 1) Input Capture Pulse Width Input Capture Pulse Period	t_{RESL} t_{ICPW} t_{ICPP}	4 125 (see note 2)	4 – –	t_{CYC} ns t_{CYC}
$\overline{\text{IRQ}}$ pulse width (edge-triggered)	t_{IPW}	125	–	ns
$\overline{\text{IRQ}}$ pulse period	t_{IPP}	(see note 2)	–	t_{CYC}
PC0-PC5 interrupt pulse width low (edge-triggered)	t_{KBIPW}	125	–	ns
PC0-PC5 interrupt pulse period	t_{KBIPP}	(see note 2)	–	t_{CYC}
OSC1 pulse width	t_{OH}, t_{OL}	90	–	ns
EEPROM byte erase time	t_{EBYTE}	–	10	ms
EEPROM block erase time	t_{BLOCK}	–	10	ms
EEPROM bulk erase time	t_{EBULK}	–	10	ms
EEPROM program time	t_{PROG}	–	10	ms

Notes:

(1) The 2-bit timer prescaler is the limiting factor in determining timer resolution.

(2) The minimum period t_{ICPPL}, t_{IPP}, or t_{KBIPP} should not be less than the number of cycles it takes to execute the interrupt service routine plus 19 t_{CYC}.

13.5 Pulse Width Modulator Timing

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

PARAMETER	SYMBOL	MINIMUM	MAXIMUM	UNIT
GPWM frame period	t_{GP}	63		t_{CYC}
GPWM step size	$t_{S.GP}$	1		t_{CYC}
RSPWM frame period (CRE=0, no PB7 reset)	t_{RSP}	127		t_{RCLK}
RSPWM step size	$t_{S.RSP}$	1		t_{RCLK}
RCLK low pulse width	t_{RCKL}	80	—	ns
RCLK high pulse width	t_{RCKH}	80	—	ns
PB7 counter reset low pulse width (CRE=1)	t_{PBRL}	80	—	ns
RCLK to RS output delay	t_{RCKRD}	—	50	ns
PB7 to RS output delay (CRE=0)	t_{PBRD}	—	90	ns

Note: t_{RCLK} is the RCLK input signal period.

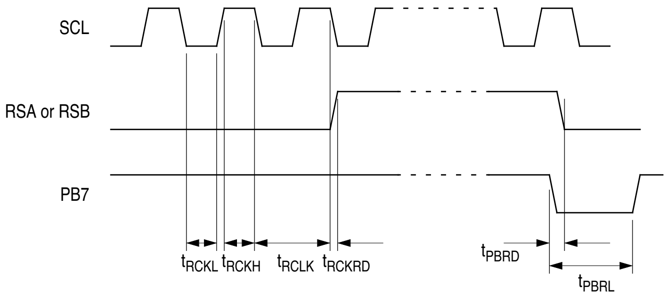


Figure 13-1 PWM Timing

13.6 M-Bus Timing

Table 13-3 M-Bus Interface Input Signal Timing

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

PARAMETER	SYMBOL	MINIMUM	MAXIMUM	UNIT
START condition hold time	$t_{HD.STA}$	2	–	t_{CYC}
Clock low period	t_{LOW}	4.7	–	t_{CYC}
Clock high period	t_{HIGH}	4	–	t_{CYC}
Data set-up time	$t_{SU.DAT}$	250	–	ns
Data hold time	$t_{HD.DAT}$	0	–	t_{CYC}
START condition set-up time (for repeated START condition only)	$t_{SU.STA}$	2	–	t_{CYC}
STOP condition set-up time	$t_{SU.STO}$	2	–	t_{CYC}

Table 13-4 M-Bus Interface Output Signal Timing

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

PARAMETER	SYMBOL	MINIMUM	MAXIMUM	UNIT
START condition hold time	$t_{HD.STA}$	8	–	t_{CYC}
Clock low period	t_{LOW}	11	–	t_{CYC}
Clock high period	t_{HIGH}	11	–	t_{CYC}
SDA/SCL rise time (see note 1)	t_R	–	1	µs
SDA/SCL fall time (see note 1)	t_F	–	300	ns
Data set-up time	$t_{SU.DAT}$	$t_{LOW} - t_{CYC}$	–	ns
Data hold time	$t_{HD.DAT}$	0	–	t_{CYC}
START condition set-up time (for repeated START condition only)	$t_{SU.STA}$	10	–	t_{CYC}
STOP condition set-up time	$t_{SU.STO}$	10	–	t_{CYC}

Note:

1. With 200pF loading on the SDA/SCL pins

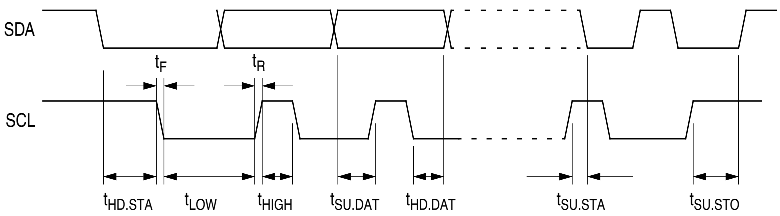


Figure 13-2 M-Bus Timing

13.7 Sync Signal Processor Timing

Table 13-5 Sync Signal Processor Timing

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

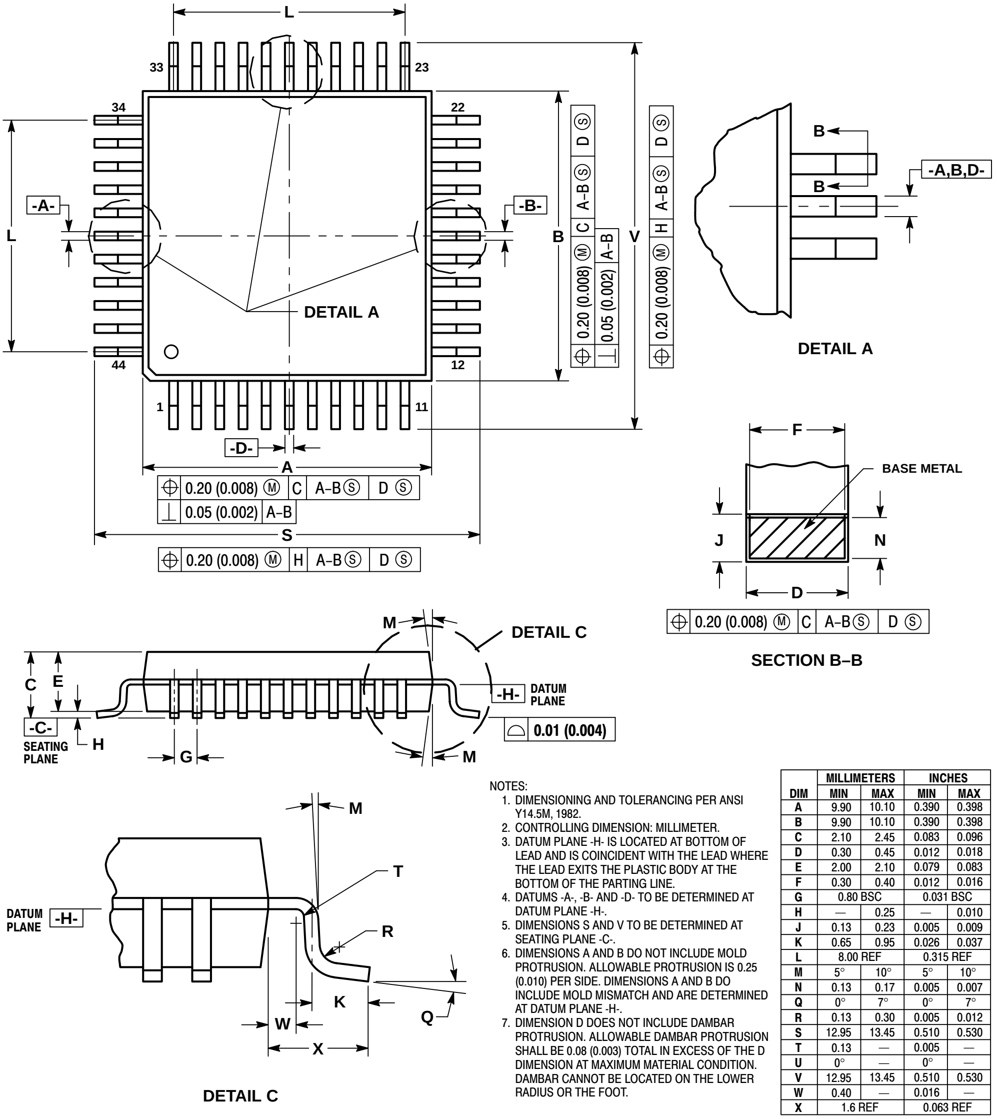
PARAMETER	SYMBOL	MINIMUM	MAXIMUM	UNIT
VSYNC input sync pulse width	$t_{VI.SP}$	2	1023	t_{CYC}
VSYNC input period	t_{VI}	15360	48127	t_{CYC}
HSYNC input sync pulse width	$t_{HI.SP}$	1.5	16	t_{CYC}
HSYNC input period	t_{HI}	—	128	t_{CYC}
Line per frame	L_{FRAME}	—	4095	LINE
VTTL output sync pulse width	$t_{VO.SP}$	6	$t_{VI.SP}$ + 6	t_{HO}
Free-running VTTL output sync pulse width (SOUT clear)	$t_{FVO.SP}$	128		t_{CYC}
Free-running VTTL output period (SOUT clear)	t_{FVO}	31360		t_{CYC}
Free-running HTTL output sync pulse width (SOUT clear)	$t_{FHO.SP}$	8		t_{CYC}
Free-running HTTL output period (SOUT clear)	t_{FHO}	35		t_{CYC}
Inserted HTTL sync pulse (INSRT cleared)	$t_{IHI.SP}$	4		t_{CYC}
Inserted HTTL period error (INSRT cleared)	$t_{IHI.ER}$	—	1	t_{CYC}
SAM sampling pulse width	t_{SAM}	1		t_{HO}
VSYNC to VTTL delay (falling edge)	t_{VVD}	—	40	ns
HSYNC to HTTL delay	t_{HHD}	—	40	ns
HSYNC to VTTL delay (rising edge)	t_{HVD}	—	40	ns
CSYNC to VTTL delay	t_{CVD}	—	60	ns
CSYNC to HTTL delay	t_{CHD}	—	60	ns

Note: t_{HO} is the HTTL output signal period.

14
MECHANICAL SPECIFICATIONS

This section provides the mechanical dimensions for the 44-pin QFP package for the MC68HC05BS8.

14.1 44-pin QFP Package



NOTES:

1. DIMENSIONING AND TOLERANCING PER ANSI Y14.5M, 1982.
2. CONTROLLING DIMENSION: MILLIMETER.
3. 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.
4. DATUMS -A-, -B- AND -D- TO BE DETERMINED AT DATUM PLANE -H-.
5. DIMENSIONS S AND V TO BE DETERMINED AT SEATING PLANE -C-.
6. DIMENSIONS A AND B DO NOT INCLUDE MOLD PROTRUSION. ALLOWABLE PROTRUSION IS 0.25 (0.010) PER SIDE. DIMENSIONS A AND B DO INCLUDE MOLD MISMATCH AND ARE DETERMINED AT DATUM PLANE -H-.
7. DIMENSION D DOES NOT INCLUDE DAMBAR PROTRUSION. ALLOWABLE DAMBAR PROTRUSION SHALL BE 0.08 (0.003) TOTAL IN EXCESS OF THE D DIMENSION AT MAXIMUM MATERIAL CONDITION. DAMBAR CANNOT BE LOCATED ON THE LOWER RADIUS OR THE FOOT.

	MILLIMETERS		INCHES	
DIM	MIN	MAX	MIN	MAX
A	9.90	10.10	0.390	0.398
B	9.90	10.10	0.390	0.398
C	2.10	2.45	0.083	0.096
D	0.30	0.45	0.012	0.018
E	2.00	2.10	0.079	0.083
F	0.30	0.40	0.012	0.016
G	0.80 BSC		0.031 BSC	
H	—	0.25	—	0.010
J	0.13	0.23	0.005	0.009
K	0.65	0.95	0.026	0.037
L	8.00 REF		0.315 REF	
M	5°	10°	5°	10°
N	0.13	0.17	0.005	0.007
Q	0°	7°	0°	7°
R	0.13	0.30	0.005	0.012
S	12.95	13.45	0.510	0.530
T	0.13	—	0.005	—
U	0°	—	0°	—
V	12.95	13.45	0.510	0.530
W	0.40	—	0.016	—
X	1.6 REF		0.063 REF	

Figure 14-1 44-pin QFP Package (Case No. 824A-01)

A
MC68HC705BS8

This appendix summarizes the differences between the MC68HC05BS8 and MC68HC705BS8. The same information can also be found in appropriate sections of the book.

The MC68HC705BS8 is an EPROM version of the MC68HC05BS8. The 10K-bytes of user ROM in the MC68HC05BS8 are replaced by 10K-bytes of user EPROM.

A.1 Features

- Functionally equivalent to MC68HC05BS8
- 10K-bytes of user EPROM
- EPROM bootstrap mode replaces Self-Check mode on the MC68HC05BS8

A.2 Memory Map

Figure A-1 shows the memory map for the MC68HC705BS8.

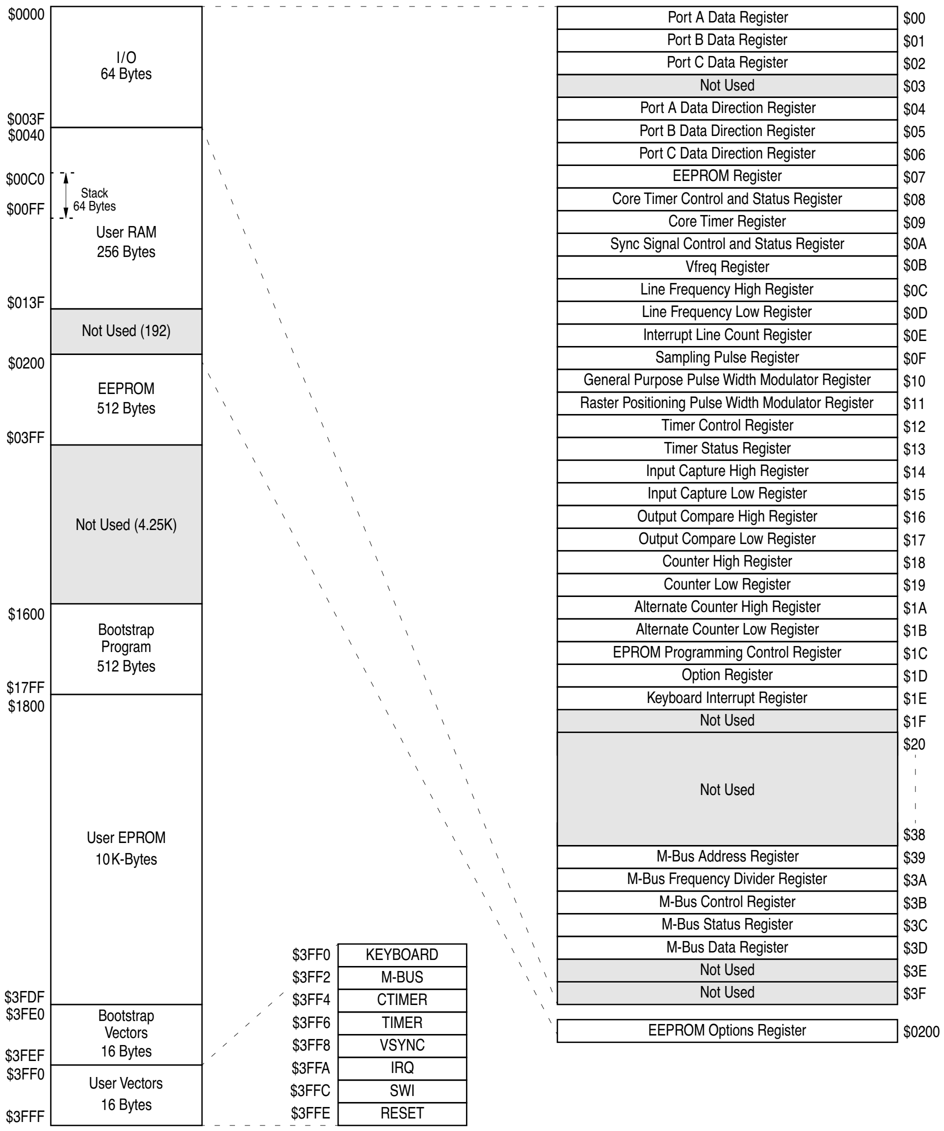

$0000
I/O
64 Bytes
$003F
$0040
$00C0
Stack
64 Bytes
$00FF
User RAM
256 Bytes
$013F
Not Used (192)
$0200
EEPROM
512 Bytes
$03FF
Not Used (4.25K)
$1600
Bootstrap
Program
512 Bytes
$17FF
$1800
User EPROM
10K-Bytes
$3FDF
$3FE0
Bootstrap
Vectors
16 Bytes
$3FEF
$3FF0
User Vectors
16 Bytes
$3FFF
$3FF0 KEYBOARD
$3FF2 M-BUS
$3FF4 CTIMER
$3FF6 TIMER
$3FF8 VSYNC
$3FFA IRQ
$3FFC SWI
$3FFE RESET
Port A Data Register $00
Port B Data Register $01
Port C Data Register $02
Not Used $03
Port A Data Direction Register $04
Port B Data Direction Register $05
Port C Data Direction Register $06
EEPROM Register $07
Core Timer Control and Status Register $08
Core Timer Register $09
Sync Signal Control and Status Register $0A
Vfreq Register $0B
Line Frequency High Register $0C
Line Frequency Low Register $0D
Interrupt Line Count Register $0E
Sampling Pulse Register $0F
General Purpose Pulse Width Modulator Register $10
Raster Positioning Pulse Width Modulator Register $11
Timer Control Register $12
Timer Status Register $13
Input Capture High Register $14
Input Capture Low Register $15
Output Compare High Register $16
Output Compare Low Register $17
Counter High Register $18
Counter Low Register $19
Alternate Counter High Register $1A
Alternate Counter Low Register $1B
EPROM Programming Control Register $1C
Option Register $1D
Keyboard Interrupt Register $1E
Not Used $1F
$20
Not Used
$38
M-Bus Address Register $39
M-Bus Frequency Divider Register $3A
M-Bus Control Register $3B
M-Bus Status Register $3C
M-Bus Data Register $3D
Not Used $3E
Not Used $3F
EEPROM Options Register $0200

Figure A-1 MC68HC705BS8 Memory Map

A.3 Modes of Operation

The MC68HC705BS8 also has two modes of operation – user mode and EPROM bootstrap mode. Table A-1 shows the conditions required to enter each mode on the rising edge of $\overline{\text{RESET}}$.

Table A-1 MC68HC705BS8 Operating Mode Entry Conditions

$\overline{\text{RESET}}$	$\overline{\text{IRQ}}$	PB6	MODE
5V ↑	V_{SS} to V_{DD}	V_{SS} to V_{DD}	USER
5V ↑	9V ↑ +9V Rising Edge*	V_{DD}	BOOTSTRAP

* Minimum hold time should be 2 clock cycles, after that it can be used as a normal $\overline{\text{IRQ}}$ function pin.

A.3.1 User Mode

The normal operating mode of the MC68HC705BS8 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.

Warning: In the MC68HC705BS8, all vectors are fetched from EPROM in user mode; therefore, the EPROM must be programmed (via the bootstrap mode) before the device is powered up in user mode.

A.3.2 Bootstrap Mode

The bootstrap mode is provided in the MC68HC705BS8 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 PB6 is at logic one. $\overline{\text{RESET}}$ must be held low for 4064 cycles after POR (power-on reset).

A.4 EPROM Programming

The Program Control register (PCR) is provided for EPROM programming. The function of the EPROM depends on the device operating mode.

A.4.1 Program Control Register (PCR)

Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
$001C	RESERVED						ELAT	PGM	uuuu uu00

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 ELATA is 1. This bit should not be set unless a programming 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 connected to the EPROM array. If ELAT ≠ 1 then PGM = 0.

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

A.4.2 EPROM Programming Sequence

Programming the EPROM of the MC68HC705BS8 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 1ms.
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 $1900 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    $1900,X      ;latch data and address
BSET   0,PCR        ;program
JSR    DELAY        ;call delay subroutine for 1ms
CLR    PCR          ;reset PCR
```

A.5 Pin Assignments

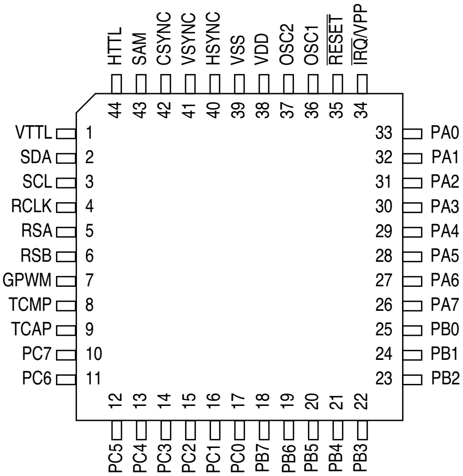


Figure A-2 Pin Assignments for 44-pin QFP package

A.6 Electrical Specifications

A.6.1 Maximum Ratings

RATINGS	SYMBOL	VALUE	UNIT
Latch-up Current for $\overline{\text{IRQ}}$	I_{LATCH}	120	mA

THIS PAGE LEFT BLANK INTENTIONALLY

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

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

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

Information in this document is provided solely to enable system and software implementers to use Freescale Semiconductor products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits or integrated circuits based on the information in this document.
Freescale Semiconductor reserves the right to make changes without further notice to any products herein. Freescale Semiconductor makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale Semiconductor assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. “Typical” parameters which may be provided in Freescale Semiconductor data sheets and/or specifications can and do vary in different applications and actual performance may vary over time. All operating parameters, including “Typicals” must be validated for each customer application by customer’s technical experts. Freescale Semiconductor does not convey any license under its patent rights nor the rights of others. Freescale Semiconductor products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the Freescale Semiconductor product could create a situation where personal injury or death may occur. Should Buyer purchase or use Freescale Semiconductor products for any such unintended or unauthorized application, Buyer shall indemnify and hold Freescale Semiconductor and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that Freescale Semiconductor was negligent regarding the design or manufacture of the part.